

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**  
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----\*\*\*\*\*-----



**BÁO CÁO**  
**MÔN HỌC: THỰC TẬP CƠ SỞ**

**Giảng viên:** Ngô Xuân Bách  
**Sinh viên:** Phan Trung Hưng  
**Mã sinh viên:** B19DCCN330  
**Lớp:** D19CQCN06-B  
**Số điện thoại:** 0364605959

**Hà Nội 2022**

## MỤC LỤC

<b>MỤC LỤC</b> .....	<b>1</b>
<b>Phần 1</b> .....	<b>2</b>
<b>Bài tập 1: Cài đặt 10 thuật toán</b> .....	<b>2</b>
1. Thuật toán Sắp xếp chọn (C++ / Java).....	2
2. Thuật toán Phân tích thừa sốNT (C++ / Java).....	3
3. Thuật toán Tìm kiếm nhị phân (C++ / Python).....	4
4. Hàng đợi ưu tiên (bài toán nối dây) (C++ / Python) .....	5
5. Thuật toán Sinh tổ hợp Ngược (C++ / C#) .....	7
6. Thuật toán Sinh hoán vị (C++ / C#).....	9
7. Thuật toán Dây con tăng dài nhất(QHĐ) (C++) .....	10
8. Thuật toán Sàng nguyên tố (C++) .....	11
9. Thuật toán Dijkstra (C++).....	13
10. Thuật toán Quân hậu (C++).....	15
<b>Phần 2</b> .....	<b>17</b>
<b>Bài tập 2: Lập trình với cơ sở dữ liệu</b> .....	<b>17</b>
<b>Chọn đề tài: Cơ sở dữ liệu với Website quản lý bài viết (Blog Online) ..</b>	<b>17</b>
1. Giới thiệu về quản lý bài viết Blog online. ....	17
2. Phân tích yêu cầu .....	17
3. Thiết kế cơ sở dữ liệu;.....	20
4. Sử dụng ngôn ngữ SQL viết truy vấn (insert, update, delete, count, thống kê).....	24
5. Xây dựng ứng dụng với CSDL đã xây dựng .....	25
<b>Phần 3</b> .....	<b>39</b>
<b>Bài tập 3: Máy tính và điều khiển thiết bị</b> .....	<b>39</b>
3.1 Kiến trúc và các thành phần cơ bản của máy tính .....	39
3.2. Nguyên lý hoạt động của hệ điều hành .....	43
<b>Phần: PHỤ LỤC</b> .....	<b>48</b>
<b>Phần Code: Cài đặt 10 thuật toán</b> .....	<b>48</b>

## Phần 1

### Bài tập 1: Cài đặt 10 thuật toán

#### 1. Thuật toán Sắp xếp chọn.

##### + ) Mô tả thuật toán:

- Thuật toán sắp xếp chọn sẽ sắp xếp một mảng bằng cách đi tìm phần tử có giá trị nhỏ nhất (giả sử với sắp xếp mảng tăng dần) trong đoạn chưa được sắp xếp và đổi cho phần tử nhỏ nhất đó với phần tử ở đầu đoạn chưa được sắp xếp.

- **Ý tưởng:** Danh sách chứa các phần tử sẽ được chia làm hai phần.

+ Phần ở bên trái là phần được sắp xếp (**sort list**)

+ Phần ở bên phải là phần chưa được sắp xếp (**unsorted list**).

=> Tại mỗi bước lặp của thuật toán, phần tử nhỏ nhất ở mảng con chưa được sắp xếp sẽ được di chuyển về đoạn đã sắp xếp.

Ban đầu chưa được sắp xếp thì phần **sorted list** sẽ trống và phần **unsorted list** sẽ chứa tất cả các phần tử ban đầu.

Phần tử nhỏ nhất trong list sẽ được chọn và trao đổi với vị trí đầu tiên trong list, tiếp đến sẽ là vị trí nhỏ thứ hai tiếp tục được trao đổi ngay sau vị trí nhỏ nhất.

##### + ) Giả mã:

```
void selectionSort(int arr[], int n) {  
    int i, j, vtriMin;  
    // Di chuyển ranh giới của mảng đã sắp xếp và chưa sắp xếp  
    for (i = 0; i < n-1; i++) {  
        // Tìm phần tử nhỏ nhất trong mảng chưa sắp xếp  
        vtriMin = i;  
        for (j = i+1; j < n; j++) {  
            if (arr[j] < arr[vtriMin])  
                vtriMin = j;  
        }  
        // Đổi chỗ phần tử nhỏ nhất với phần tử đầu tiên  
        swap(arr[vtriMin], arr[i]);  
    }  
}
```

##### + ) Ứng dụng

- Sắp xếp chọn là một trong những thuật toán đơn giản. Nó thực hiện công việc so sánh các phần tử trong danh sách. Sau đó sắp xếp và trả về một danh sách theo chiều tăng hoặc giảm dần.

- Thuật toán sắp xếp chọn được sử dụng trong các trường hợp:

- Mảng có ít phần tử
- Chi phí hoán đổi không thành vấn đề
- Kiểm tra tất cả các phần tử là bắt buộc

**+) Độ phức tạp:**  $O(n^2)$

**+) Code:** dưới phần phụ lục bằng ngôn ngữ **C++** và **Java**

- Tham khảo: <https://freetuts.net/thuat-toan-sap-xep-chon-selection-sort-2931.html>

## **2. Thuật toán Phân tích thừa số NT**

**+) Mô tả thuật toán:**

- Tích các thừa số nguyên tố chính là phép nhân giữa các số với nhau trong đó tất cả các số đều là **số nguyên tố**. Ví dụ:  $10 = 2 * 5$ . Trong đó 2 và 5 là các số nguyên tố.

- Mục tiêu của bài toán là: Nhập vào một số nguyên dương N, hãy phân tích số N thành tích các thừa số nguyên tố (chính là phần phía sau dấu bằng).

- **Ý tưởng:** chỉ cần thực hiện chia số N cho các số nguyên tố trong đoạn  $[2; N]$ . Với mỗi số nguyên tố đó, đếm số lần mà số N chia hết. Tất nhiên, sau mỗi lần chia cho số i, số N của chúng ta sẽ giảm đi i lần. Chừng nào N chưa bằng 1, ta tiếp tục quá trình chia.

⇒ **Cải tiến thuật toán:**

- Chỉ cần thực hiện chia số N trong khoảng từ  $[2; \sqrt{n}]$ , và chia như ý tưởng trên. Vì nếu số N chia hết cho 1 số trong khoảng từ  $[2; \sqrt{n}]$ , thì cũng chia hết cho 1 số trong khoảng từ  $[\sqrt{n}, n]$ .

- Khi số N chia tới  $\sqrt{n}$ , N sau khi chia nếu  $= 1$  thì giống như thuật toán ban đầu, nếu  $N \neq 1$ , thì lúc đó N chắc chắn là 1 số nguyên tố, và kết thúc thuật toán.

- Mục đích: giảm độ phức tạp tính toán

**+) Cài đặt:**

```
void ThuaSoNTTo(long long n){
    //chạy từ 2 -> sqrt(n): số i nhỏ nhất mà n chia hết sẽ là số ngto
    for(long i=2;i<=sqrt(n);i++) {
        int dem=0;
        while(n%i==0){
            dem++;
            n/=i;
        }
        //nếu dem != 0 thì in ra các thừa số ngto + số mũ (dem)
        if(dem!=0){
            cout<<i<<"^"<<dem<<" x ";
        }
    }
    //kết quả n sau vòng for = 1 hoặc = 1 số ngto
    if(n!=1)
        cout<<n<<"^1";    //chỉ cần in số ngto cuối cùng
}
```

#### + ) Ứng dụng

Vấn đề phân tích thừa số nguyên tố cũng khá được quan tâm trong lập trình thi đấu, và nó còn có một số ứng dụng khác trong số học.

Nhiều giao thức mã hoá khóa dựa trên việc phân tích các số nguyên lớn này để giảm thời gian tính toán.

+ ) **Độ phức tạp:**  $O(n \log(n))$

+ ) **Code:** dưới phần phụ lục bằng ngôn ngữ **C++** và **Java**

Tham khảo: <https://cachhoc.net/2017/05/31/phan-tich-thua-nguyen/>

### 3. Thuật toán Tìm kiếm nhị phân

#### + ) Mô tả thuật toán:

- Một cách tìm kiếm đơn giản hơn rất nhiều so với thuật toán tìm kiếm tuần tự (linear search) có độ phức tạp lớn lên tới  $O(n)$ , không khả thi để thực hiện tìm kiếm trên một mảng lớn. Để giải quyết bài toán này chúng ta sẽ sử dụng thuật toán **tìm kiếm nhị phân (binary search)**, đây là thuật toán phổ biến để tìm kiếm vị trí một phần tử trong một mảng đã sắp xếp.

- Cho một danh sách `arr[]` đã được sắp xếp gồm  $n$  phần tử, viết một hàm đưa ra vị trí của phần tử `x` trong mảng.

- **Ý tưởng:** Để triển khai thuật toán này hãy chắc chắn rằng mảng đã được sắp xếp. Do tính chất mảng đã sắp xếp, công việc tìm kiếm phần tử `x` có thể triển khai như sau:

1. Xét đoạn mảng `arr[left...right]` cần tìm kiếm phần tử `x`. Ta so sánh `x` với phần tử ở vị trí giữa của mảng (`mid = (left + right)/2`). Nếu:
2. Nếu phần tử `arr[mid] = x`. Kết luận và thoát chương trình.

3. Nếu  $\text{arr}[\text{mid}] < x$ . Chỉ thực hiện tìm kiếm trên đoạn  $\text{arr}[\text{mid}+1 \dots \text{right}]$ .
4. Nếu  $\text{arr}[\text{mid}] > x$ . Chỉ thực hiện tìm kiếm trên đoạn  $\text{arr}[\text{left} \dots \text{mid}-1]$ .
5. Nếu ko tìm thấy mid, trả về -1. Kết quả trong mảng ko chứa giá trị x cần tìm.

**+) Giải mã:**

```
int TKNhiPhan(int arr[], int n, int xCanTim){
    int left = 0, right = n-1;
    while(left <= right){
        //mid là vị trí chia đôi của mảng
        int mid = (left + right)/2;
        if(xCanTim == arr[mid])
            return mid;
        else if(xCanTim < arr[mid])
            right = mid-1;
        else
            left = mid+1;
    }
    return -1;
}
```

**+) Ứng dụng**

Thuật toán tìm kiếm nhị phân với độ phức tạp tính toán nhanh, được sử dụng rất nhiều trong thực tế cho phép tìm kiếm vị trí của một phần tử trong một mảng đã được sắp xếp.

Nếu việc tìm kiếm tuần tự từ trên xuống dưới mất nhiều thời gian, thì tìm kiếm nhị phân sẽ tìm rất nhanh và giảm số lần tìm kiếm trên phần tử.

Thuật toán hay được sử dụng trong kế toán: tìm kiếm thông tin người dùng, hoặc giá trị nào đó với số lượng lớn.

**+) Độ phức tạp:** tốt nhất là  $O(1)$ , trung bình cũng là  $O(\log_2 n)$ , trường hợp xấu nhất là  $O(n)$

**+) Code:** dưới phần phụ lục bằng ngôn ngữ C++ và Python

Tham khảo: <https://freetuts.net/thuat-toan-tim-kiem-nhi-phan-2634.html>

#### **4. Hàng đợi ưu tiên (bài toán nối dây)**

**+) Mô tả thuật toán:**

- Phát biểu bài toán nối dây:

- + Cho n dây với chiều dài khác nhau. Cần nối các dây lại với nhau thành một dây.
- + Chi phí nối hai dây lại với nhau được tính bằng tổng độ dài hai dây.

+ Nhiệm vụ của bài toán là tìm cách nối các dây lại với nhau thành một dây sao cho **chi phí nối** các dây lại với nhau là **ít nhất**.

- **Ví dụ:** Số lượng dây là 4; Độ dài dây  $L[] = \{4, 3, 2, 6\}$

- Dây số 3 nối với dây số 2  $\Rightarrow$  3 dây với độ dài 4, 5, 6.
- Dây độ dài 4 nối với dây độ dài 5  $\Rightarrow$  2 dây với độ dài 6, 9.
- Nối hai dây còn lại  $6+9=15$ .

$\Rightarrow$  Tổng chi phí nhỏ nhất là  $5 + 9 + 15 = 29$ .

- Hàng đợi ưu tiên hay **Priority Queue** là một loại hàng đợi đặc biệt, trong đó mỗi phần tử được liên kết với một mức độ ưu tiên và được thực thi theo mức độ ưu tiên đó.

- Sự khác biệt giữa Priority Queue và hàng đợi thông thường: Trong một hàng đợi thông thường, quy tắc vào trước ra trước được thực hiện, trong khi đó, trong hàng đợi ưu tiên, các giá trị được lấy dựa trên cơ sở mức độ ưu tiên. Phần tử có mức ưu tiên cao nhất sẽ ra trước.

- **Ý tưởng:** sử dụng hàng đợi ưu tiên **Priority Queue** để giải quyết bài toán. Ở bài toán này, độ ưu tiên là **tăng dần**, mục đích để **pop** ra 2 phần tử đầu tiên của Priority Queue. Cộng tổng giá trị 2 phần tử này và **push** lại vào Priority Queue. Khi push vào, hàng đợi ưu tiên sẽ lại tự động sắp xếp các phần tử trong queue theo thứ tự tăng dần.

+)**Giải mã:**

❖ **Input:** n: số lượng dây;  $L[]$  : chi phí nối dây.

❖ **Output:** Chi phí nối dây nhỏ nhất.

```
+ Bước 1: Tạo priority_queue pq là hàng đợi ưu tiên tăng dần,  
|         | Lưu trữ tất cả độ dài n dây trong mảng.  
+ Bước 2: (Lặp).  
|         | OPT = 0;  
|         | while (pq.size() > 1) {  
|         |     First = pq.top(); pq.pop();  
|         |     Second = pq.top(); pq.pop();  
|         |     OPT = First + Second;  
|         |     pq.push(First + Second);  
|         | }  
+ Bước 3: (Trả lại kết quả tổng nối dây nhỏ nhất).  
|         | Return(OPT);
```

+)**Ứng dụng hàng đợi ưu tiên**

- Ứng dụng hàng đợi ưu tiên để lập lịch công việc. Khi ta có một hàng đợi với N công việc, mỗi công việc đều có độ ưu tiên của riêng nó. Nếu công việc có độ ưu tiên cao nhất sẽ được hoàn thành trước và xóa bỏ khỏi hàng đợi, hoặc có thể thêm các công việc vào danh sách để quản lý tốt.

- Tìm chi phí nổi dây nhỏ nhất
- Để cân bằng tải và xử lý các ngắt trong hệ điều hành.
- Để triển khai trong việc nén dữ liệu Huffman.

+) **Độ phức tạp:** Mảng hàng đợi ưu tiên: Pop  $O(1)$ ; Top  $O(1)$ ; Push  $O(n)$

+) **Code:** dưới phân phụ lục bằng ngôn ngữ **C++** và **Python**

Tham khảo: [slide chương 4.Tham lam \(Cấu trúc dữ liệu giải thuật - thầy N.M.Son\)](#)

## 5. Thuật toán Sinh tổ hợp Ngược

+) **Mô tả thuật toán:**

- Phương pháp sinh có thể áp dụng để giải bài toán liệt kê tổ hợp đặt ra nếu như hai điều kiện sau thỏa mãn:

- Có thể xác định được một thứ tự trên tập các cấu hình tổ hợp cần liệt kê. Từ đó có thể biết được cấu hình đầu tiên và cấu hình cuối cùng trong thứ tự đó.
- Xây dựng được thuật toán từ một cấu hình chưa phải cấu hình cuối, sinh ra được cấu hình kế tiếp nó.

- Mô tả chung của thuật toán sinh

<Xây dựng cấu hình đầu tiên>;

(repeat)

<Đưa ra cấu hình đang có>;

<Từ cấu hình đang có sinh ra cấu hình kế tiếp nếu còn>;

(until <hết cấu hình>;)

- **Thuật toán sinh tổ hợp.** Bài toán này giúp ta liệt kê được tất cả các tổ hợp chập K của N phần tử. Khi liệt kê, ta sẽ ưu tiên hiển thị các phần tử theo tổ hợp đã được sắp xếp theo kiểu từ điển. Như vậy nếu ta đang có một dãy x đại diện cho một tập con, nếu x là cấu hình kết thúc có nghĩa là tất cả các phần tử trong x đều đã đạt tới giới hạn trên thì quá trình sinh kết thúc, nếu không thì ta phải sinh ra một dãy x mới tăng dần thỏa mãn **vừa đủ lớn hơn dãy cũ** theo nghĩa không có một tập con k phần tử nào chen giữa chúng khi sắp thứ tự từ điển.

=> **Thuật toán sinh tổ hợp ngược** được cải tiến lên từ thuật toán sinh tổ hợp chập K của N phần tử. Chỉ thay đổi **cấu hình hết thức** và việc **giảm** các phần tử thay vì tăng như sinh tổ hợp xuôi

- **Ví dụ:** sinh ngược chập 3 của 5 phần tử ta được lần lượt là: 345, 245, 235, 234 145, 135, 134, 125, 124, 123

- **Ý tưởng:**



- Khởi tạo cấu hình ban đầu: mảng arr chứa các k phần tử: k, k-1, k-2, ..., 1.
- Tìm vị trí mà tại đó phần tử [i] chưa đạt cấu hình cuối (  $a[i] \neq 1$  &&  $a[k] = k$  ).
- Nếu tìm thấy, giảm [i]--; và các phần tử sau nó tính từ trái sang phải giảm dần:  $[k] = n$ ,  $[k-1] = n-1$ , ... Lần lượt cho tới khi  $arr[k] = k$  thì dừng.
- Nếu không tìm thấy thì đã đạt cấu hình cuối, nghĩa là tất cả các phần tử trong x đều đã đạt tới giới hạn trên thì quá trình sinh kết thúc (đánh dấu = false để thoát).

**+) Cài đặt:**

```
void SinhToHopNGUOC(int arr[], int n, int k){
    int i=k, tmp=n;
    //lặp cho tới khi gặp cấu hình kết thúc là arr[1] = 1
    while(arr[i] == arr[i-1]+1){
        i--;
    }
    //dừng khi lặp hết các phần tử hoặc gặp cấu hình kết thúc arr[1] = 1
    if(i==1 && a[1]==1)
        dd=false;
    else{
        //nếu ko thì [i] giảm dần
        arr[i]--;
        //các phần tử sau giảm dần từ phải qua trái
        for(int j=k; j>i; j--){
            arr[j] = tmp;
            tmp--;
        }
    }
}
```

**+) Ứng dụng**

+ Đối với những bài toán xác suất thống kê, tìm các cấu hình, hoán vị, tổ hợp, chỉnh hợp, từ ba bài toán trên ta có thể tùy biến và sử dụng được để góp phần xử lý cho những bài toán lớn và phức tạp hơn.

Theo wikipedia: [https://vi.wikipedia.org/wiki/Toán\\_học\\_tổ\\_hợp](https://vi.wikipedia.org/wiki/Toán_học_tổ_hợp)

+ Ứng dụng nghiên cứu về các cấu hình kết hợp các phần tử của một tập hợp có hữu hạn phần tử. Các cấu hình đó là các hoán vị, chỉnh hợp, tổ hợp,... các phần tử của một tập hợp.

+ Giải quyết các bài toán Toán học tổ hợp

+ Liên quan đến cả khía cạnh giải quyết vấn đề lẫn xây dựng cơ sở lý thuyết

**+) Code:** dưới phần phụ lục bằng ngôn ngữ C++ và C#

Tham khảo: <https://v1study.com/giai-thuat-va-lap-trinh-phuong-phap-sinh-generation.html#gsc.tab=0>

## **6. Thuật toán Sinh hoán vị**

### **+) Mô tả thuật toán:**

- Mô tả chung của thuật toán sinh

<Xây dựng cấu hình đầu tiên>;

(repeat)

<Đưa ra cấu hình đang có>;

<Từ cấu hình đang có sinh ra cấu hình kế tiếp nếu còn>;

(until <hết cấu hình>;)

- Hoán vị sẽ sinh ra phải lớn hơn hoán vị hiện tại, hơn thế nữa phải là hoán vị vừa đủ lớn hơn hoán vị hiện tại theo nghĩa không thể có một hoán vị nào khác chen giữa chúng khi sắp thứ tự.

- Như vậy hoán vị đầu tiên sẽ là  $(1, 2, \dots, n)$ . Hoán vị cuối cùng là  $(n, n-1, \dots, 1)$ .

- **Ví dụ:** hoán vị 3 phần tử 123 lần lượt là: 123, 132, 213, 231, 312, 321

hoán vị 5 phần tử 12345 lần lượt là: 12345, 12354, 12435, 12453, 12534, ... 54321

### **- Ý tưởng:**

- Bước 1: Tìm vị trí  $k$  lớn nhất thỏa  $A_k < A_{k+1}$ , nếu không tìm được thì đây là hoán vị cuối cùng, dừng thuật toán.
- Bước 2: Tìm vị trí  $L$  lớn nhất thỏa  $A_k < A_L$
- Bước 3: Đổi chỗ 2 phần tử  $A_k$  và  $A_L$ .
- Bước 4: Đảo ngược mảng trong đoạn  $(A_{k+1}, A_{k+2}, \dots, A_n)$ .

### **+) Cài đặt:**

```
void SinhHoanVi(int arr[], int n){
    int i = n-1;
    //cấu hình kết thúc khi arr[1] = n
    //vòng lặp so sánh từ phải qua trái tìm vị trí đứng trước bé hơn đứng sau
    while(arr[i] > arr[i+1])
        i--;
    //dừng khi là phần tử cuối cùng hoặc arr[1] = n
    if(i==0)
        dd=0;
    else{
        int j=n;
        //tìm vị trí bên phải arr[i] lớn hơn nhỏ nhất so với arr[i]
        //để đổi chỗ với arr[i]
        while(arr[i] > arr[j])
            j--;
        swap(arr[i], arr[j]);
        int l=i+1, r=n;
        //đảo ngược mảng trong đoạn bên phải so với arr[i]
        while(l<r){
            swap(arr[l], arr[r]);
            l++; r--;
        }
    }
}
```

#### + ) Ứng dụng

+ Đối với những bài toán xác suất thống kê, tìm các cấu hình, hoán vị, tổ hợp, chỉnh hợp, từ ba bài toán trên ta có thể tùy biến và sử dụng được để góp phần xử lý cho những bài toán lớn và phức tạp hơn

+ Thực hiện số lần đổi chỗ và tìm các trạng thái đổi chỗ trong các bài toán thực tế như đổi chỗ ngôi học sinh, đổi chỗ các thành phần, đổi chỗ hoặc kiểm tra đổi chỗ các mã gen với nhiều trạng thái.

+ ) **Code:** dưới phân phụ lục bằng ngôn ngữ C++ và C#

Tham khảo: <https://vietcodes.github.io/algo/permute>

Tham khảo: [Slide cấu trúc dữ liệu và giải thuật thầy N.M.Son](#)

## 7. Thuật toán Dãy con tăng dài nhất(QHĐ)

#### + ) Mô tả thuật toán:

- Dãy con tăng dài nhất là một trong những bài toán QHĐ kinh điển, tìm ra dãy con nào thỏa 2 điều kiện: tăng dần và có độ dài là lớn nhất..

- Phát biểu bài toán:

Cho dãy số A có N phần tử, bài toán yêu cầu tìm dãy con dài nhất của dãy A sao cho phần tử sau của dãy con luôn lớn hơn phần tử trước. Dãy con của một dãy số là dãy có được sau khi loại bớt một số phần tử, các phần tử khác giữ nguyên vị trí.

Nói cách khác, dãy con tăng của A là một dãy  $A(i_1), A(i_2), \dots, A(i_k)$  thỏa mãn  $i_1 < i_2 < \dots < i_k$  và  $A(i_1) < A(i_2) < \dots < A(i_k)$ .

- **Ví dụ:** mảng có 7 phần tử: 3 4 -1 0 6 2 3

=> Dãy con tăng dài nhất là gồm 4 phần tử: -1 0 2 3

- **Ý tưởng:**

Công thức truy hồi:

+ Giả sử với  $i$  chạy từ  $n$  về 0, ta cần tính  $L[i]$ : độ dài dãy con tăng dài nhất bắt đầu tại  $a_i$ .  $L[i]$  được tính trong điều kiện  $L[i + 1], L[i + 2], \dots, L[n + 1]$ .

Cách cài đặt:

+ Gọi  $F(i)$  là dãy con tăng dài nhất kết thúc ở  $A(i)$ , ta có công thức tính:

$$F(1)=1; \quad F(i)=\max\{F(j)+1\}$$

+ Với  $j$  thỏa mãn  $1 \leq j < i$  và  $A(j) < A(i)$ .

=> Kết quả bài toán là  $\max\{F\}$ .

+) **Giải mã:**

```
int result = 1;
for (int i=1; i<=n; i++) {
    f[i] = 0;
    for (int j=0; j<i; j++) if (a[j] < a[i]) {
        f[i] = max(f[i], f[j] + 1);
    }
    result = max(result, f[i]);
}
```

+) **Ứng dụng**

- Tìm các chuỗi tăng dài nhất trong 1 danh sách theo 1 tiêu chí nào đó.
- Bài toán thực tế như cách xếp hàng các học sinh theo chiều cao hoặc tìm 1 dãy theo tiêu chí trong danh sách lẫn lộn.

+) **Độ phức tạp:**  $O(N^2)$

+) **Code:** dưới phần phụ lục bằng ngôn ngữ C++

Tham khảo: <https://vietcodes.github.io/algo/lis>

## 8. Thuật toán Sàng nguyên tố

+) **Mô tả thuật toán:**

- Số nguyên tố là số nguyên dương có duy nhất 2 ước phân biệt là 1 và chính nó. Số nguyên tố nhỏ nhất là số 2.

- **Sàng nguyên tố:** dùng để tìm các số nguyên tố nhỏ hơn hoặc bằng số nguyên NN nào đó. Dựa theo lý thuyết về số nguyên tố: Một số nguyên tố là số chỉ có 2 ước là 1 và chính nó. Do vậy, nếu ta xác định được số  $x$  là số nguyên tố, ta có thể kết luận mọi số chia hết cho  $x$  đều không phải số nguyên tố. Do đó ta đã loại bỏ được rất nhiều số mà không cần kiểm tra.

Nguyên lý hoạt động của sàng là vào mỗi lần duyệt, ta chọn một số nguyên tố và loại ra khỏi sàng tất cả các bội của số nguyên tố đó mà lớn hơn số đó. Sau khi duyệt xong, các số còn lại trong sàng đều là số nguyên tố.

- **Ví dụ:** Số 2 là số nguyên tố  $\Rightarrow$  các số 4, 6, 8, 10, ... không phải số nguyên tố.

Số 3 là số nguyên tố  $\Rightarrow$  các số 9, 15, 21, ... không phải số nguyên tố. (Do 6, 12, 18 đã bị loại ở số 2)

- **Ý tưởng:** Thuật toán sàng nguyên tố thực hiện như sau:

+ Bước 1: Tạo mảng đánh dấu cho tất cả các phần tử từ 2 đến  $N$  và mặc định tất cả đều là số nguyên tố

+ Bước 2: Xét số đầu tiên tìm được là số nguyên tố – giả sử  $x$ , đánh dấu tất cả các ước của  $x$ :  $2x, 3x, 4x, \dots$  trong đoạn  $[x, N]$  không phải số nguyên tố.

+ Bước 3: Tìm số tiếp theo được đánh dấu là số nguyên tố trong  $[x, N]$ . Nếu không còn số nào, thoát chương trình. Nếu còn, gán nó bằng  $x$  và lặp lại bước 2.

+ Bước 4: Khi kết thúc giải thuật, các số không bị đánh dấu là các số nguyên tố

+) **Cài đặt:**

```
void SangNguyenTo(int n){
    bool sang[n+1];
    //đánh dấu tất cả các ptử mảng = true là số nguyên tố, loại bỏ ptử thứ 0 và 1
    for(int i=0; i<=n; i++){
        sang[i]=true;
        sang[0]=sang[1]=false;
    }

    //xét lần lượt các ptu mảng được đánh dấu.
    for(i=2; i<=sqrt(n); i++){
        if(sang[i]==true){
            //nếu ptu hiện tại = true là số nguyên tố
            //thì tất cả ptu 2*i, 3*i, ... ko là số nguyên tố, đánh dấu là false
            for(int j=2; j<=n/i; j++){
                sang[i*j]=false;
            }
        }
    }

    //sau khi sàng kết thúc, các ptu = true sẽ là các số nguyên tố
    for(i=0; i<=n; i++){
        if(sang[i]==true)
            cout<<i<<" ";
    }
}
```

#### +)**Ứng dụng**

- Ứng dụng của sàng số nguyên tố trong bài toán phân tích ra thừa số nguyên tố
- Nó còn có thể được sử dụng để kiểm tra một số nguyên nhỏ hơn hoặc bằng N hay không.
- Nhanh chóng liệt kê các số nguyên tố hoặc tìm 1 số nguyên tố trong 1 khoảng nhất định.

#### +)**Độ phức tạp:**

- Số lần lặp của vòng lặp trong là:
  - Khi  $i=2$ , vòng lặp trong lặp  $N/2$  lần.
  - Khi  $i=3$ , vòng lặp trong lặp  $N/3$  lần.
  - ... tiếp tục cho tới khi gặp N
- Độ phức tạp tổng:  $N.(1/2+1/3+1/5+...)=O(N\log N)$

#### +)**Code:** dưới phần phụ lục bằng ngôn ngữ C++

Tham khảo: <https://vnoi.info/wiki/translate/he/Number-Theory-2.md>

### 9. Thuật toán Dijkstra

#### +)**Mô tả thuật toán:**

- Các bài toán về tìm đường đi ngắn nhất và biến tướng của nó luôn xuất hiện rất nhiều trong các cuộc thi lập trình thi đấu bởi sự đa dạng trong cách đưa ra đề bài và sử dụng. Một trong những thuật toán tìm đường đi ngắn nhất được sử dụng phổ biến đó là thuật toán Dijkstra.

- Dijkstra là **thuật toán đường đi ngắn nhất**. Thuật toán Dijkstra được sử dụng để tìm khoảng cách ngắn nhất của tất cả các nút từ nút bắt đầu đã cho. Nó tạo ra cây đường đi ngắn nhất từ một nút nguồn một cách hợp lý, bằng cách tiếp tục thêm các nút một cách tham lam sao cho tại mọi điểm mỗi nút trong cây có khoảng cách tối thiểu từ nút bắt đầu đã cho.

- Thuật toán Dijkstra là một **tham lam** phương pháp sử dụng một thực tế toán học rất đơn giản để chọn một nút ở mỗi bước.

Trong trường hợp đồ thị  $G=(V,E,w)$  có trọng số trên các cạnh không âm, ta có thuật toán Dijkstra để tìm đường đi ngắn nhất từ đỉnh xuất phát s tới các đỉnh khác của đồ thị.

#### - **Ý tưởng của thuật toán:**

Ta có mảng  $kc[u]$  là khoảng cách ngắn nhất từ đỉnh s tới đỉnh u trên đồ thị. Ban đầu  $kc[s] = 0$ , các giá trị khác bằng dương vô cực. Ta sẽ lấy đỉnh u có  $kc[u]$  nhỏ nhất vào thời điểm hiện tại, và sử dụng khoảng cách của nó để cập nhật khoảng cách ngắn nhất

của các đỉnh xung quanh. Với một đỉnh  $u$  bất kì, vì nó được cập nhật bởi các đường đi ngắn nhất của các đỉnh xung quanh nó, nên bản thân đường đi của nó cũng là ngắn nhất.

Các bước thực hiện chính:

1. Đánh dấu đỉnh ban đầu (đỉnh nguồn) là 0 và các đỉnh còn lại là vô cùng.
2. Gọi đỉnh chưa xét với giá trị đánh dấu nhỏ nhất là  $C$  (current node).
3. Với mỗi đỉnh kề  $N$  (neighbour) với đỉnh  $C$ : Cộng giá trị đang đánh dấu của đỉnh  $C$  với trọng số của cạnh nối đỉnh  $C$  với đỉnh  $N$ . Nếu được kết quả nhỏ hơn giá trị đang đánh dấu ở đỉnh  $N$  ta cập nhật giá trị đánh dấu cho đỉnh  $N$  mới.
4. Đánh dấu đỉnh  $C$  đã xét xong.
5. Nếu vẫn còn đỉnh chưa xét, lặp lại bước 2.

**+) Giải mã:**

```
void Dijkstra(Graph, source):
    dist[source] := 0                // Khoảng cách từ nguồn đến nguồn được đặt thành 0
    for each vertex v in Graph:      // vòng lặp để khởi tạo
        if v ≠ source
            dist[v] := infinity      // Hàm khoảng cách từ nguồn đến mỗi nút, đặt thành vô cùng
            add v to Q               // Tất cả các nút ban đầu trong Q

    while Q is not empty:            // Khi ở vòng lặp chính
        v = vertex in Q with min dist[v] // Trong lần chạy đầu tiên, đỉnh này là nút nguồn
        remove v from Q

        for each neighbor u of v:    // Nơi người hàng xóm u vẫn chưa được xóa khỏi Q
            dist[u] = min(dist[u], dist[v] + length(v, u))

    return dist[]
end function
```

**+) Ứng dụng:** Một số ứng dụng của thuật toán Dijkstra trong thực tế:

- Được sử dụng trong định tuyến với một chương trình con trong các thuật toán đồ thị hay trong công nghệ Hệ thống định vị toàn cầu (GPS).
- Tìm đường đi ngắn nhất trên bản đồ.
- Ứng dụng trong mạng xã hội.
- Ứng dụng trong hệ thống thông tin di động
- Ứng dụng trong hàng không
- Ứng dụng cho tìm điểm hạ cánh cho máy bay ngắn nhất,...

**+) Độ phức tạp:** Thuật toán Dijkstra bình thường sẽ có độ phức tạp là  $O(n^2+m)$ , do ta phải duyệt  $n$  lần (đối với  $n$  đỉnh), mỗi lần duyệt lại phải duyệt qua  $n$  đỉnh để tìm đỉnh có  $kc[u]$  nhỏ nhất.

**+) Code:** dưới phần phụ lục bằng ngôn ngữ C++

Tham khảo: [https://vi.tutorialcup.com/interview/algorithm/dijkstra-algorithm.htm#Time\\_complexity\\_Onn](https://vi.tutorialcup.com/interview/algorithm/dijkstra-algorithm.htm#Time_complexity_Onn)

Tham khảo: <https://viblo.asia/p/thuat-toan-dijkstra-va-ung-dung-aWj53zgQl6m>

## 10. Thuật toán Quân hậu

### +) Mô tả thuật toán:

- Bài toán 8 quân hậu là một ví dụ nổi tiếng về việc dùng các phương pháp thử và phương pháp quay lui. Một quân hậu trên bàn cờ có thể di chuyển theo hàng ngang, cột dọc và 2 đường chéo.

- Phát biểu bài toán: Cho một bàn cờ có kích thước  $N \times N$  ( $N \geq 1$ ), Bạn có thể đặt đúng  $N$  quân hậu lên bàn cờ (mỗi ô chỉ chứa tối đa một quân hậu), hãy đưa ra cách đặt  $N$  quân hậu sao cho không có 2 quân hậu nào ăn được nhau, nói cách khác là trên mỗi hàng, một cột, mỗi đường chéo của bàn cờ chỉ chứa tối đa một quân hậu.

### - Ý tưởng:

- Đầu tiên ta đặt quân hậu thứ nhất vào các cột trên hàng 1 ( có  $n$  cách đặt ).
- Thử đặt quân hậu 2 vào từng cột ở hàng 2 sao cho không bị quân hậu 1 khống chế. Với mỗi vị trí của quân hậu này ta lại thử đặt quân hậu thứ ba vào các cột sao cho không bị các quân hậu trước khống chế.
- Sau khi đặt xong quân hậu thứ tám thì in ra một cách đặt.
- Cách kiểm tra một ô vuông có nằm trong tầm ngắm của các quân hậu trước đó hay không:
  - Sử dụng mảng boolean  $c$  để đánh dấu các cột của bàn cờ ( $c[i] = \text{true}$  nếu trên cột  $i$  chưa đặt quân hậu nào)
  - Sử dụng mảng bool  $c1$  để đánh dấu các đường chéo song song với đường chéo chính của bàn cờ ( $c[i - j + N - 1] = \text{true}$ , nghĩa là đường chéo đi qua ô  $(i, j)$  và song song với đường chéo chính chưa được đặt quân hậu nào.
  - Sử dụng mảng bool  $c2$  để đánh dấu các đường chéo song song với đường chéo phụ của bàn cờ ( $c[i + j - 2] = \text{true}$ , nghĩa là đường chéo đi qua ô  $(i, j)$  và song song với đường chéo phụ chưa được đặt quân hậu nào.

=> Sau tìm xong vị trí của quân hậu thứ  $N$  thì ta output trường hợp này và tiếp tục trường hợp khác.

### +) Cài đặt:



```
//hàm kiểm tra các trường hợp đặt Hậu có thỏa mãn ko, trả về true nếu thỏa mãn
bool Check(int c, int r){
    for(int i=1; i<c ;i++){
        if( a[i] == r || abs(a[i]-r) == abs(i-c) )
            return false;
    }
    return true;
}

//hàm quay lui quân hậu
void Try(int i,int n){
    for(int j=1; j<=n; j++){ //đặt Ậu vào các cột từ 1 đến 8
        if(Check(i, j)){ //nếu có thể thì đặt Hậu vào
            a[i] = j;
            if(i==n) //kiểm tra đủ n ô thì output kết quả đó
                HienThi(n);
            Try(i+1, n); //tiếp tục quay lui các trường hợp khác
        }
    }
}
```

### +) Ứng dụng

- Trong lập trình AI, bài toán N quân Hậu là một trong những bài toán cơ bản, thường được dùng để đánh giá sức mạnh của một thuật toán.

- Ngoài ra bài toán N quân Hậu cũng được ứng dụng trong nhiều lĩnh vực khác. Theo wikipedia

+) **Độ phức tạp:**  $O(n^n)$ .

+) **Code:** dưới phần phụ lục bằng ngôn ngữ C++

Tham khảo: <https://giaithuatlaptrinh.github.io/Quay-lui/>

Tham khảo: <https://nguyenvanhieu.vn/bai-toan-xep-hau/>

## **Phần 2**

### **Bài tập 2: Lập trình với cơ sở dữ liệu**

#### **Chọn đề tài: Cơ sở dữ liệu với Website quản lý bài viết (Blog Online)**

### **1. Giới thiệu về quản lý bài viết Blog online.**

#### **+) Blog online là gì?**

- **Blog** là một website thông tin riêng hoặc nhật ký trực tuyến, với cách trình bày các bài viết mới nhất được đưa lên đầu. Người viết blog có thể là cá nhân hay một nhóm nhỏ, thể hiện cái nhìn chủ quan của họ về một chủ đề nhất định, viết về những điều họ thích.

- Về cơ bản, blog là một trang web được sử dụng để liệt kê các bài đăng trên blog theo thứ tự thời gian.

- Một blog có nội dung gần đây nhất được hiển thị đầu tiên, tiếp theo là nội dung đã cập nhật trước đó.

- Trong một blog có nhiều chuyên mục và bài viết được viết bởi một số lượng lớn các tác giả khác nhau cung cấp thông tin.

#### **+) Hệ thống quản lý bài viết blog online**

- Hệ thống quản lý bài viết đã phát triển cho phép người dùng tạo blog của riêng họ, nơi người dùng có thể công bố ý kiến của riêng họ.

- Trong hệ thống này có các quản trị viên chăm sóc Blog.

- Nếu bất kỳ ai muốn đăng bài trước tiên họ sẽ phải tạo một blog mới cũng như một tài khoản trang web.

- Các bài viết có thể được xác minh bởi các quản trị viên.

- Có danh sách danh mục nơi các bài viết có thể được lưu theo danh mục liên quan.

- Mọi người có thể thích, bình luận và chia sẻ các bài viết

### **2. Phân tích yêu cầu**

#### **2.1. Mô tả hệ thống quản lý bài viết**

- Cấu trúc của blog sẽ có thanh tiêu đề, menu, đăng nhập, đăng kí thông tin.

- Trang chủ chứa tất cả những bài viết mới nhất, những comment của mọi người.

- Sidebar thường là để thêm thông tin về mạng xã hội, những bài viết đặc biệt hoặc những bài viết được xem nhiều nhất.

- Mỗi người dùng khi đăng kí hoặc đăng nhập đều có thể tạo các bài viết mới cho cá nhân và mọi người cùng xem. Đồng thời có thể comment, bình luận, thả reaction cho các bài viết của mình cũng như của người khác.

- Người dùng viết bài đăng nào, sẽ được phép thêm, sửa, xóa bài đăng của người đó. Người dùng comment bình luận ở bài nào, sẽ được phép thêm, sửa, xóa bình luận của người dùng đó.

- Nếu là quản trị viên có thể quản lí người dùng blog như thêm, sửa, xóa, cập nhật thông tin người dùng. Quản trị viên cũng có thể thêm, sửa, xóa, cập nhật thông tin các bài viết blog trên trang chủ. Quản trị viên cũng có thể thêm sửa xóa, cập nhật tất cả bình luận của tất cả người dùng trên trang chủ.

## **2.2. Mục đích và yêu cầu:**

- Xây dựng một hệ thống website để tất cả mọi người dùng có thể chia sẻ những thông tin hữu ích, những trải nghiệm hoặc cảm xúc cá nhân lên một hệ thống, qua đó kết nối người dùng thông qua các bài viết đó.

- **Với người dùng:** phân quyền thành 2 nhóm chính là **Admin** và **User**.

### **\* Admin:**

+ Admin có thể quản lí số lượng, thông tin User khi đăng nhập đăng kí  
+ Có thể thêm, sửa, xóa thông tin User trong hệ thống website.  
+ Có thể quản lí và biết tất cả các bài viết của các cá nhân trên trang chủ của hệ thống.

+ Có thể thêm, sửa, xóa, cho hiển thị, cập nhật tất cả những bài viết, sắp xếp các bài viết theo mục đích của quản trị viên.

+ Tính năng bình luận cũng được Admin quản lý, có thể sửa, xóa của bất kì User trên bất kì bài viết nào.

### **\* User**

+ User có thể truy cập hệ thống website mọi lúc mọi nơi, và đọc, xem các bài viết mới nhất từ cộng đồng và các thông tin của quản trị viên.

+ Mọi người dùng đều có thể tạo mới tài khoản đăng kí, đăng nhập vào hệ thống để sử dụng hệ thống.

+ Mọi người dùng đều có thể tạo bài viết trên hệ thống (nếu ko bị admin thu hồi quyền truy cập)

+ Có thể xem tất cả các bài viết của hệ thống được gắn ở trang chủ.

+ Có thể bình luận tất cả các bài viết trên trang chủ. Và có thể chỉnh sửa, xóa bình luận của chính người dùng đó.

- **Với hệ thống website:**

- + Hiện thị tất cả danh sách người dùng cho quản trị viên
- + Hiện thị tất cả các bài viết cho mọi người có link vào hệ thống
- + Chỉ khi đăng nhập hoặc đăng kí mới được có thêm các chức năng như: tạo mới bài viết, bình luận bài viết, thêm xóa, sửa bài viết và bình luận. Lọc các bài viết theo yêu cầu.

### **2.3. Các trang và chức năng của hệ thống:**

#### **- Trang chủ:**

- + Chứa tất cả các bài viết của các user và admin, gồm có tiêu đề, ngày giờ đăng, thông tin người dùng đăng, nội dung bài đăng. Theo **thứ tự mới nhất**.
- + Khi Click vào từng bài viết, người dùng đăng bài đó có thể chỉnh sửa, xóa bài đăng cũng như trả lời các bình luận.
- + Khi Click vào từng bài viết, người dùng khác có thể bình luận bài viết đó, và cũng có thể chỉnh sửa các bình luận của chính người dùng đó.

#### **- Bình luận:**

- + Người dùng được bình luận qua câu văn hoặc văn bản được lưu vào hệ thống và chỉ định ở bài viết đó.

#### **- Chỉnh sửa bài đăng:**

- + Admin hoặc người đăng bài đó sẽ được chỉnh sửa bài viết đó theo nhu cầu.

#### **- Tạo bài viết mới:**

- + Mọi người dùng khi đăng nhập có thể tạo 1 bài viết mới và được lưu vào hệ thống, hiển thị ở trang chủ cùng với tên người đăng.
- + 1 bài viết mới bao gồm: tiêu đề, nội dung, và hình ảnh kèm theo bài viết, và thông tin người dùng sẽ được gắn theo.

#### **- Lọc bài viết:**

- + Mọi người dùng khi đăng nhập đều có thể dùng chức năng này
- + Người dùng có thể lọc các bài viết thông qua các từ khóa nhập vào theo chiều mới nhất hoặc cũ nhất.

#### **- Quản lý người dùng:**

- + Chỉ có Admin mới có chức năng này
- + Admin có thể lọc tất cả người dùng thông qua các từ khóa, các tên nhu cầu vào theo chiều từ điển xuôi và ngược của bảng chữ cái.
- + Khi Click vào 1 user, admin sẽ nhận được tất cả thông tin của user đó, và có thể cấp quyền cho người dùng đó bằng cách chặn hoặc bỏ chặn dùng hệ thống.

#### **- Đăng kí:**

+ Mọi người dùng đều có thể đăng kí 1 tài khoản để truy cập vào hệ thống và cho cả những lần đăng nhập sau.

**- Đăng nhập:**

+ Mọi người dùng sau 1 lần đăng kí sẽ có thể truy cập vào hệ thống để sử dụng

+ Mỗi lần đăng nhập hệ thống sẽ giữ phiên đăng nhập của người dùng đó, và sử dụng hệ thống, cho tới khi đăng xuất khỏi hệ thống (thông qua session)

**3. Thiết kế cơ sở dữ liệu;**

**Sử dụng MySQL tạo database : csdlblog**

**+) Bảng User chứa:**

**-id: INT (PK)**

**-user\_name: VARCHAR(45)**

**-role: VARCHAR(45)**

**-password: VARCHAR(255)**

**-deactive: INT**

**+) Bảng Post chứa:**

**-id: INT**

**-user\_id: INT**

**-title: VARCHAR(2000)**

**-img: VARCHAR(1000)**

**-content\_post: VARCHAR(5000)**

**-created\_time: TIMESTAMP(3)**

**-updated\_time: TIMESTAMP(3)**

**+) Bảng Comment chứa:**

**Id: INT**

**user\_id: INT**

**user\_name: VARCHAR(45)**

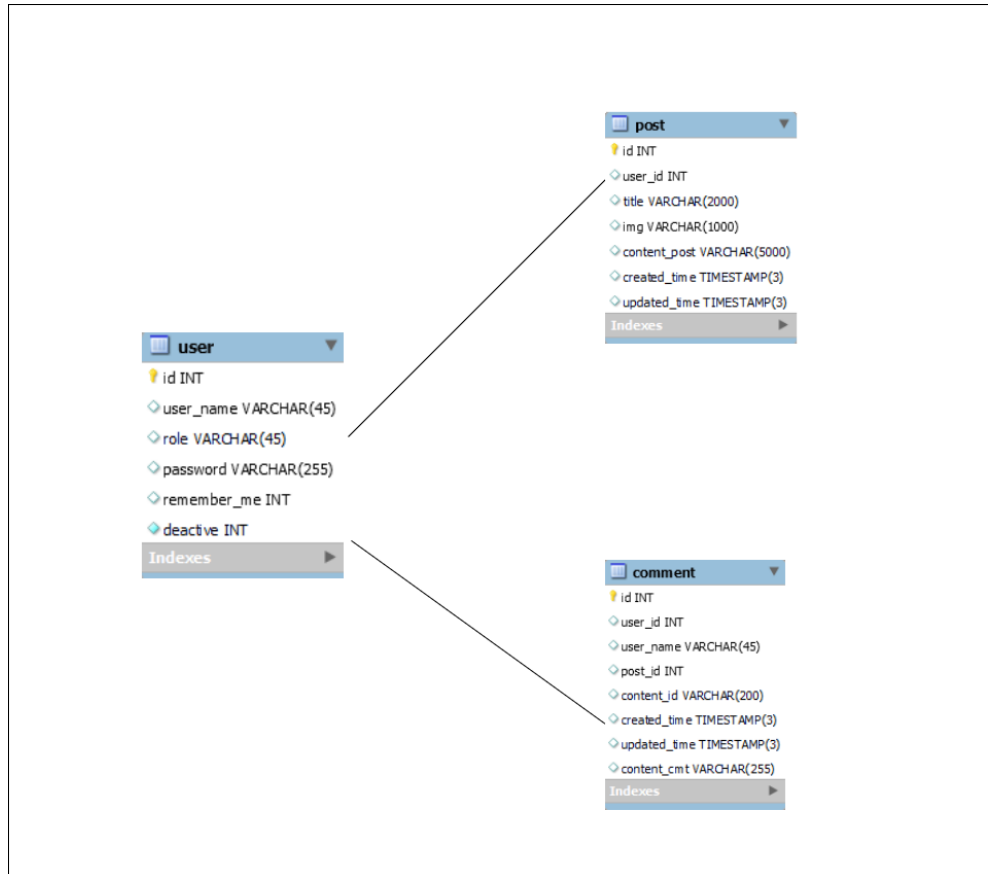
**post\_id: INT**

**content\_id: VARCHAR(200)**

**created\_time: TIMESTAMP(3)**

**updated\_time: TIMESTAMP(3)**

**content\_cmt: VARCHAR(1000)**



**Các thực thể:**

**+) user (id, user\_name, role, password, remember\_me, deactive)**

**+) post (id, user\_id, title, img, content\_post, created\_time, updated\_time)**

**+) comment (id, user\_id, user\_name, post\_id, content\_id, created\_time, updated\_time, content\_cmt)**

**\*) Hình ảnh một số dữ liệu sẵn các bảng trong MySQL:**

**+) Bảng user**

user - Table

Table Name:  Schema: **csdlblog**

Charset/Collation:   Engine: **InnoDB**

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
user_name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
role	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'ROLE_USER'
password	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
remember_me	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'
deactivate	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'

user post comment

Limit to 1000 rows

1 • `SELECT * FROM csdlblog.user;`

Result Grid

	id	user_name	role	password	remember_me	deactive
▶	2	hung123	ROLE_ADMIN	12345	1	0
	4	pth123	ROLE_USER	12345	1	0
	6	pth123456	ROLE_USER	12345	1	0
	7	hung1702	ROLE_USER	12345	1	0
	8	hung123456789	ROLE_USER	12345	1	0
	9	hungpt	ROLE_USER	12345	1	0
	11	admin	ROLE_ADMIN	12345	1	0
	12	hungpt	ROLE_USER	12345	1	0
*	NULL	NULL	NULL	NULL	NULL	NULL

+ ) Bảng post

post - Table

Table Name:  Schema: **csdlblog**

Charset/Collation:   Engine: **InnoDB**

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
user_id	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
title	VARCHAR(2000)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
img	VARCHAR(1000)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
content_post	VARCHAR(5000)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
created_time	TIMESTAMP(3)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP(3)
updated_time	TIMESTAMP(3)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP(3)

user post comment

Limit to 1000 rows

1 • SELECT \* FROM csdlblog.post;

Result Grid

	id	user_id	title	img	content_post	created_time	updated_time
▶	1	2	Chuyên gia công nghệ chia sẻ về công việc lập t...	<a href="https://www.jobhopin.com/blog/wp-content/upl...">https://www.jobhopin.com/blog/wp-content/upl...</a>	Tìm hiểu về công việc lập trình viên Lập trình vi...	2022-05-28 21:58:51.731	2022-05-28 21:58:51.731
	2	2	Học lập trình mất bao lâu?	<a href="https://image.shutterstock.com/image-photo/y...">https://image.shutterstock.com/image-photo/y...</a>	Thật ra thì rất khó để có thể xác định chính xác ...	2022-05-29 13:41:18.510	2022-05-29 13:41:18.510
	4	2	6 yếu tố để tạo nên lập trình viên thành công	<a href="https://media.1nguoiduatin.vn/media/nguyen-...">https://media.1nguoiduatin.vn/media/nguyen-...</a>	Ngoài năm chắc các kiến thức liên quan đến JAV...	2022-05-29 13:42:32.923	2022-05-29 13:42:32.923
	5	9	Sinh viên IT mua máy tính nào tốt?	<a href="https://cdn.tgdd.vn/Files/2021/07/29/1371593...">https://cdn.tgdd.vn/Files/2021/07/29/1371593...</a>	Có rất nhiều bạn tân sinh viên hỏi về việc mua ...	2022-05-29 13:43:55.398	2022-05-29 13:43:55.398
	6	9	Mình đã học lập trình như thế nào?	<a href="https://cafedev.vn/wp-content/uploads/2020/0...">https://cafedev.vn/wp-content/uploads/2020/0...</a>	Mình đã học lập trình như thế nào? Trong video ...	2022-05-29 13:45:21.529	2022-05-29 13:45:21.529
	7	2	Lập trình viên cần học những gì?	<a href="https://phattrienphanmem123az.com/wp-conte...">https://phattrienphanmem123az.com/wp-conte...</a>	Một trong những câu hỏi được nhiều bạn sinh vi...	2022-05-29 13:47:01.917	2022-05-29 13:47:01.917
	8	2	Bắt đầu viết blog, viết blog được và mất gì?	<a href="https://laptrinhcuocsong.com/images/read-blog...">https://laptrinhcuocsong.com/images/read-blog...</a>	Hiện nay có rất nhiều các lập trình viên viết blog...	2022-05-29 13:48:30.960	2022-05-29 13:48:30.960
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## +) Bảng comment

comment - Table

Table Name:  Schema: **csdlblog**

Charset/Collation:   Engine: **InnoDB**

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
user_id	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
user_name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
post_id	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
content_id	VARCHAR(200)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
created_time	TIMESTAMP(3)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP(3)
updated_time	TIMESTAMP(3)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP(3)
content_cmt	VARCHAR(1000)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL



The screenshot shows a database management tool interface. At the top, there are tabs for 'user', 'post', and 'comment'. The 'comment' tab is selected. Below the tabs, there is a toolbar with various icons and a text area containing the SQL query: `1 • SELECT * FROM csdlblog.comment;`. Below the query, there is a 'Result Grid' showing the results of the query. The grid has columns for 'id', 'user\_id', 'user\_name', 'post\_id', 'content\_id', 'created\_time', 'updated\_time', and 'content\_cmt'. The results are as follows:

id	user_id	user_name	post_id	content_id	created_time	updated_time	content_cmt
1	7	hung1702	1	NULL	2022-05-29 14:16:19.685	2022-05-29 14:16:19.685	Bài viết hay quá. Cảm ơn bạn !
2	7	hung1702	2	NULL	2022-05-29 14:16:56.440	2022-05-29 14:16:56.440	Bài viết của bạn rất bổ ích. Thanks
3	11	admin	1	NULL	2022-05-29 14:17:38.972	2022-05-29 14:17:38.972	Cảm ơn bạn đã chia sẻ.
4	2	hung123	1	NULL	2022-06-01 05:21:15.795	2022-06-01 05:21:15.795	Chúc bạn 1 ngày tốt lành!
5	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#### 4. Sử dụng ngôn ngữ SQL viết truy vấn (insert, update, delete, count, thống kê)

##### + ) Lệnh truy vấn với bảng user:

###### - Lấy ra các user:

```
SELECT * FROM csdlblog.user;
```

###### - Lấy ra 1 user với user\_name:

```
SELECT * FROM csdlblog.user WHERE user_name = 'hung123';
```

###### - Thêm mới 1 user

```
INSERT INTO csdlblog.user (user_name, password) VALUES ('phanhung', '12345');
```

###### - Cập nhật 1 user với user\_name:

```
UPDATE csdlblog.user SET user_name = 'hungpt' WHERE (id = 12);
```

###### - Xóa 1 user

```
DELETE from user WHERE id = 12;
```

##### + ) Lệnh truy vấn với bảng post:

###### - Lấy ra tất cả bài viết:

```
SELECT * FROM csdlblog.post;
```

###### - Lấy ra một bài viết theo id:

```
SELECT * FROM csdlblog.post WHERE id = 5;
```

###### - Chén 1 bài post vào bảng:

```
INSERT INTO csdlblog.post (user_id, title, img, content_post) VALUES(10, 'Titleabc', 'abc.png', 'This is contend of title');
```

###### - Cập nhật 1 bài viết:

UPDATE post set title= 'Titleabc', img ='abc.png', content\_post = 'This is contend of title' WHERE id = 6;

**- Xóa 1 bài viết:**

DELETE from post WHERE id = ?

**+) Lệnh truy vấn với bảng comment**

**- Lấy ra các comment bài viết theo id bài viết**

SELECT \* FROM csdlblog.comment WHERE post\_id = id

**- Thêm 1 comment theo id bài viết.**

INSERT INTO csdlblog.comment (user\_id, user\_name, post\_id, content\_cmt) VALUES (?, ?, ?, ?)

## **5. Xây dựng ứng dụng với CSDL đã xây dựng**

### **5.1. Các công nghệ sử dụng:**

+ **Backend:** ngôn ngữ **Java**, các công nghệ **Java Spring Boot, Jdbc Template.**

+ **Frontend:** **Html, css**, thư viện của **thymeleaf**

+ **Database:** sử dụng kết nối qua **MySQL**

+ **IDE:** **Intellij IDEA – JetBrains**

+ Chạy trên **localhost** cổng 9000: **http://localhost:9000/**

=> Sử dụng mô hình **MVC (model – view - controller)** để xây dựng website

### **5.2. Xây dựng website**

**\*) Phần Data:**

**+) Tạo các file Repository thao tác đến Database sử dụng Jdbctemplate**

+ **CommentRepository:** hàm lấy ra danh sách các bình luận và bài viết và hàm lưu bình luận vào database

+ **PostRepository:** lấy ra tất cả bài viết, lọc các bài viết hoặc lấy ra theo yêu cầu, xóa, sửa và cập nhật bài viết

+ **UserRepository:** lấy ra thông tin người dùng, tìm kiếm người dùng, thêm, sửa, xóa người dùng hoặc chặn người dùng, cập nhật người dùng

**+) Tạo các file Model đại diện cho từng model trong ứng dụng**

+ **model User**: chứa các thuộc tính của Bảng user trong database như tên đăng nhập, mật khẩu, phân quyền role cho user và admin

+ **modle Post**: chứa các thuộc tính và các dữ liệu cho bài viết như tiêu đề, nội dung, link ảnh

+ **modle Comment**: chứa các thuộc tính về người bình luận, ngày giờ và nội dung bình luận, cập nhật bình luận

**\*) Phần Controller:**

**+) HomeController**

- **GetMapping /** : trả về giao diện trang chính hiển thị danh sách các bài viết, các chức năng của hệ thống

+) **PostController**: Chỉ người dùng tạo ra bài viết hoặc admin của hệ thống mới có quyền chỉnh sửa và xóa bài viết

- **GetMapping /post/{id}** : Trả về giao diện hiển thị ra thông tin chi tiết của bài viết với id tương ứng

- **PostMapping /createNewPost** : Lưu thông tin bài viết người dùng vừa đăng tải vào database

- **GetMapping /editPost/{id}** : Trả về form chỉnh sửa thông tin chi tiết bài viết với id tương ứng

- **GetMapping /deletePost/{id}** : Xóa bài viết với id tương ứng nếu là chủ nhân hoặc admin

- **PostMapping /filter** : Tìm kiếm và hiển thị các bài viết phù hợp với yêu cầu tìm kiếm của người dùng

+) **AdminController** : Dành riêng cho người dùng có role là Admin, có thể quản lý người dùng

- **GetMapping /manage** : Trả về giao diện quản lý người dùng bao gồm danh sách người dùng và form thống kê tìm kiếm người dùng

- **GetMapping /user/{id}** : Trả về giao diện hiển thị chi tiết thông tin về người dùng có id tương ứng

- **GetMapping /banUser/{id}**: Cấm người dùng có id tương ứng khỏi hệ thống

- **PostMapping /searchUser** : Tìm kiếm người dùng tương ứng với form search

**+) CommentController**

- **GetMapping /comment/{id}** : Lấy ra các comment tương ứng với bài post có id tương ứng

- **PostMapping /comment** : đăng tải comment tương ứng lên và lưu vào database nếu chưa đăng nhập đẩy sang trang login để người dùng đăng nhập với có quyền comment

**+) LoginController**

- **PostMapping /signin** : xác thực thông tin username, password của người dùng nếu chính xác lưu thông tin người đang đăng nhập vào session và trả về giao diện home

- **GetMapping /logout** : Thoát người dùng đang đăng nhập khỏi hệ thống và xóa thông tin người dùng khỏi session

**+) RegisterController**

- **PostMapping /register** : Tạo mới người dùng sau khi xác thực thông tin phù hợp thì lưu xuống database và lưu thông tin người dùng vào session và trả về giao diện home

**\*) Tạo các templates bằng file html để hiển thị giao diện:**

- **home.html** : để vào trang home giao diện hiện ra list các bài post của hệ thống
- **signup.html** : để hiển thị form đăng ký người dùng
- **login.html** : để hiển thị form đăng nhập người dùng
- **filterPost.html** : để hiển thị trang tìm kiếm filter về các bài post phù hợp
- **formComment.html** : để hiển thị trang tạo comment mới
- **formPost.html** : để hiển thị trang tạo post mới
- **manage.html** : hiển thị ra trang tìm kiếm filter về các user phù hợp
- **userDetail.html** : hiển thị ra thông tin chi tiết về người dùng

**+) Các components html :**

- **footer.html** : phần footer của trang web
- **header.html** : phần header của trang web
- **posts.html**: phần list ra các bài viết
- **users.html**: phần list ra các user

**\*) Trong application.properties**

chỉ định url của database: **url=jdbc:mysql://localhost/csdlblog**

chỉ định username của db: **username=root**

chỉ định password của db: **password=admin**

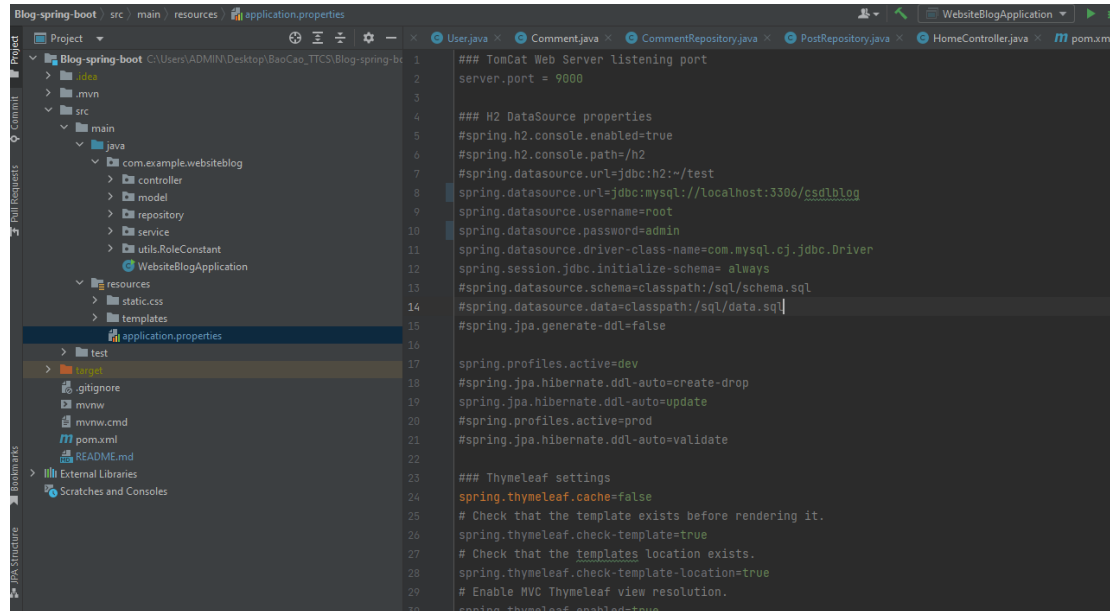
**\*) Trong pom.xml:**

thêm **dependencies** tương ứng.

### 5.3. Một số hình ảnh trong file code

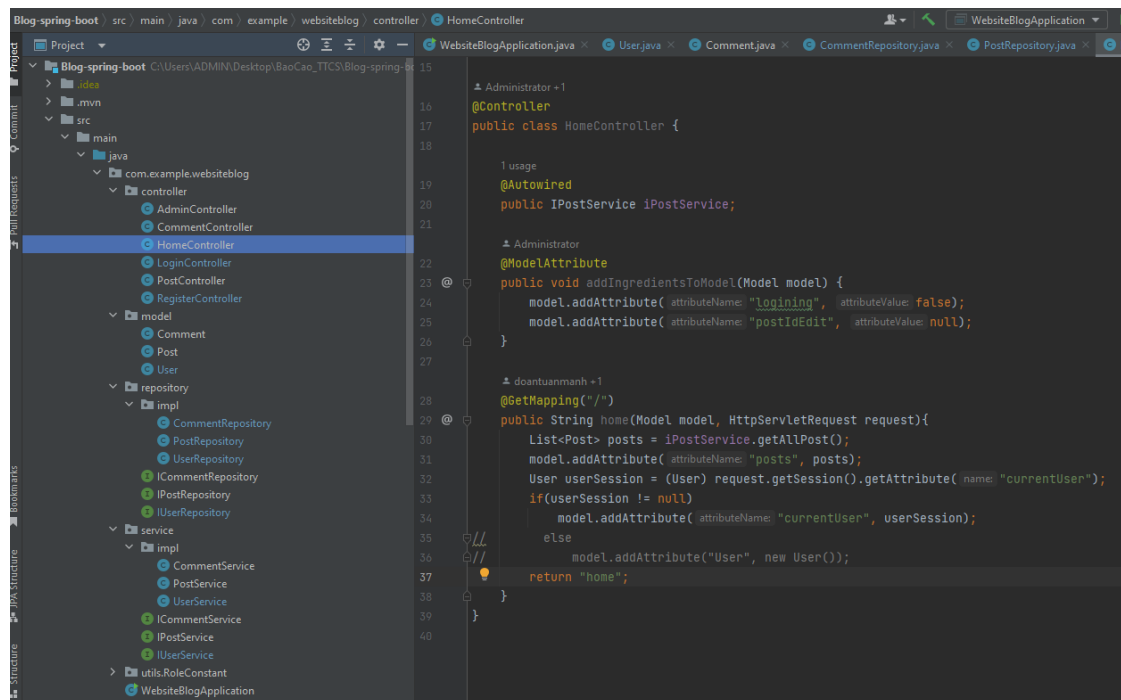
#### 5.3.1. Hình ảnh demo 1 số file phần Back end

- Tổng thể các file code của website và file pom.xml: cài đặt kết nối database.



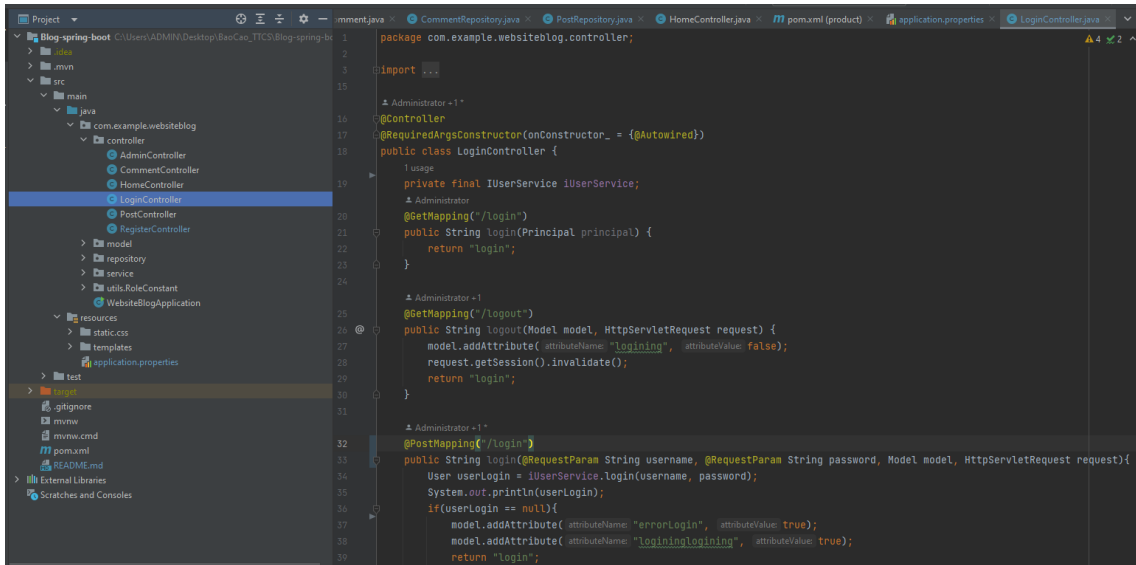
```
1  ## TomCat Web Server listening port
2  server.port = 9000
3
4  ## H2 DataSource properties
5  #spring.h2.console.enabled=true
6  #spring.h2.console.path=/h2
7  #spring.datasource.url=jdbc:h2:~/test
8  spring.datasource.url=jdbc:mysql://localhost:3306/csdblog
9  spring.datasource.username=root
10 spring.datasource.password=admin
11 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
12 spring.session.jdbc.initialize-schema= always
13 #spring.datasource.schema=classpath:/sql/schema.sql
14 #spring.datasource.data=classpath:/sql/data.sql
15 #spring.jpa.generate-ddl=false
16
17 spring.profiles.active=dev
18 #spring.jpa.hibernate.ddl-auto=create-drop
19 spring.jpa.hibernate.ddl-auto=update
20 #spring.profiles.active=prod
21 #spring.jpa.hibernate.ddl-auto=validate
22
23 ## Thymeleaf settings
24 spring.thymeleaf.cache=false
25 # Check that the template exists before rendering it.
26 spring.thymeleaf.check-template=true
27 # Check that the templates location exists.
28 spring.thymeleaf.check-template-location=true
29 # Enable MVC Thymeleaf view resolution.
30 spring.thymeleaf.enabled=true
```

- **HomeController (demo)**

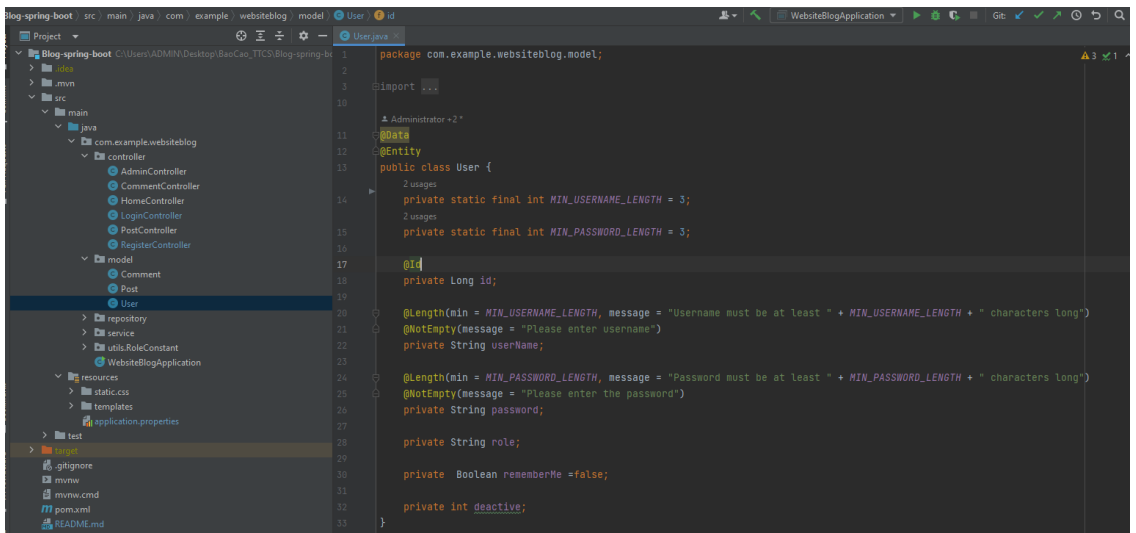


```
15
16  @Controller
17  public class HomeController {
18
19      // usage
20      @Autowired
21      public IPostService iPostService;
22
23      // Administrator
24      @ModelAttribute
25      public void addIngredientsToModel(Model model) {
26          model.addAttribute( attributeName: "logining", attributeValue: false);
27          model.addAttribute( attributeName: "postIdEdit", attributeValue: null);
28      }
29
30      // doantuanmanh +1
31      @GetMapping("/")
32      public String home(Model model, HttpServletRequest request){
33          List<Post> posts = iPostService.getAllPost();
34          model.addAttribute( attributeName: "posts", posts);
35          User userSession = (User) request.getSession().getAttribute( name: "currentUser");
36          if(userSession != null)
37              model.addAttribute( attributeName: "currentUser", userSession);
38          else
39              model.addAttribute("User", new User());
40          return "home";
41      }
42  }
```

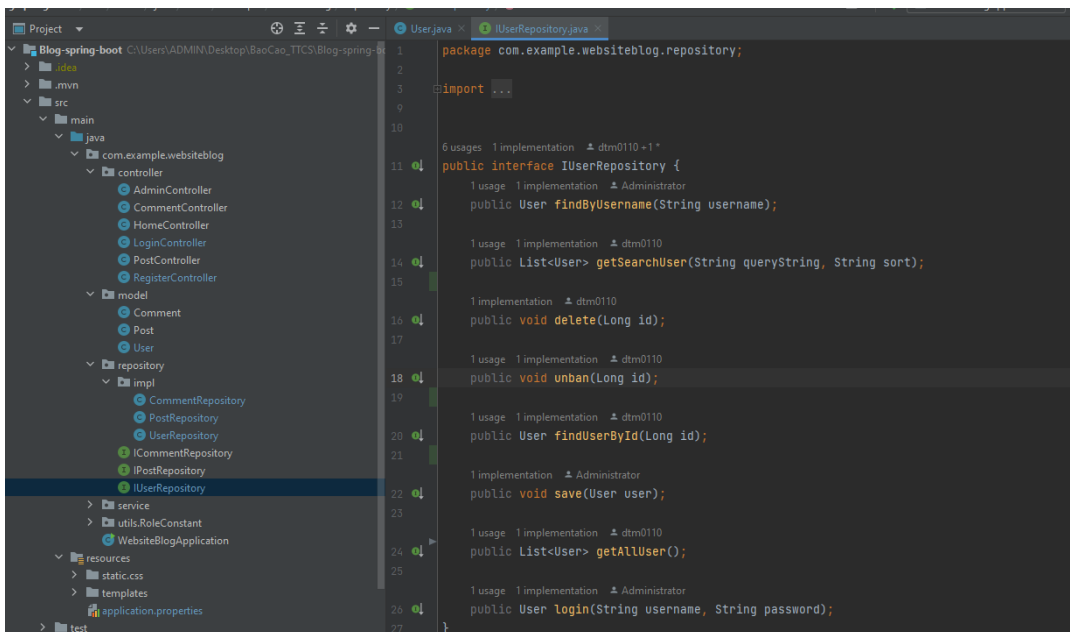
- **LoginController (demo)**



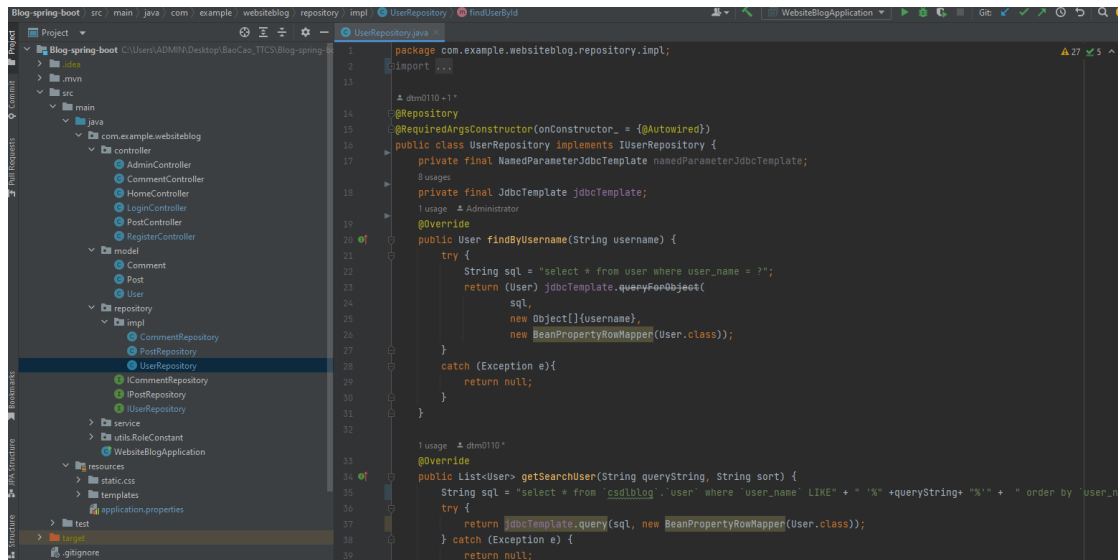
- **Model: class user**



- repository: interface của userRepository:

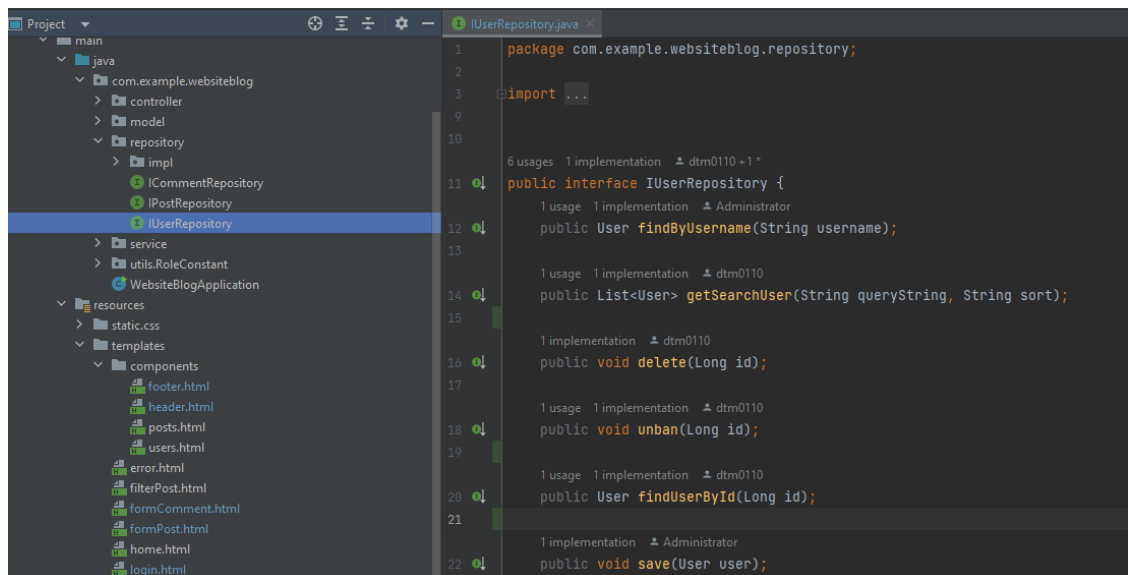


## - UserRepository: thực thi interface IUserRepository: chứa các lệnh truy vấn



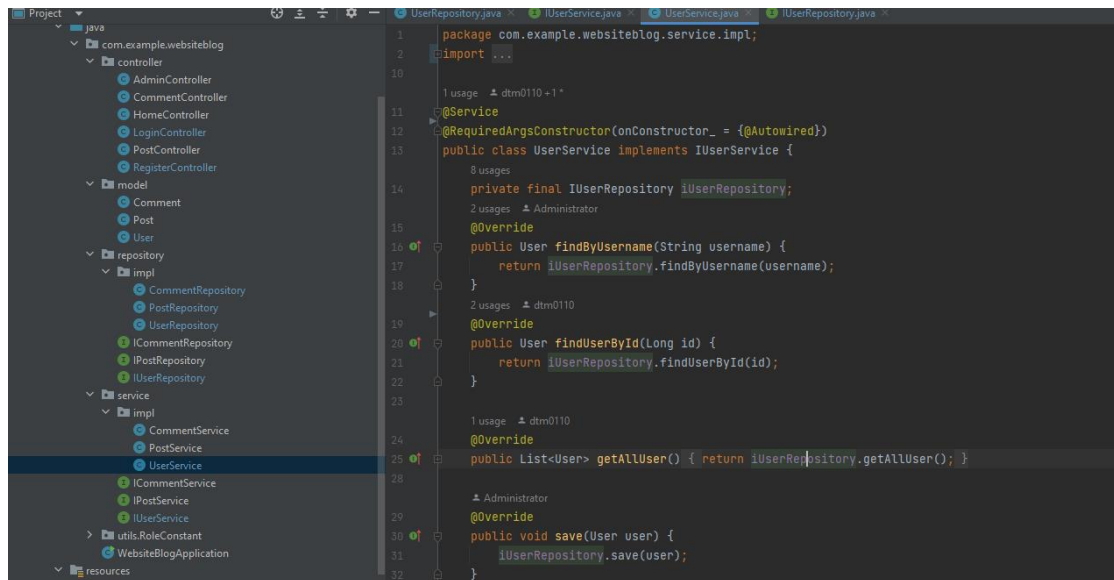
```
1 package com.example.websiteblog.repository.impl;
2 import ...
3
4 @Autowired
5 @Repository
6 @RequiredArgsConstructor(onConstructor = @Autowired)
7 public class UserRepository implements IUserRepository {
8     private final NamedParameterJdbcTemplate namedParameterJdbcTemplate;
9     private final JdbcTemplate jdbcTemplate;
10    @Usage 1 Administrator
11    @Override
12    public User findByUsername(String username) {
13        try {
14            String sql = "select * from user where user_name = ?";
15            return (User) jdbcTemplate.queryForObject(
16                sql,
17                new Object[]{username},
18                new BeanPropertyRowMapper(User.class));
19        } catch (Exception e) {
20            return null;
21        }
22    }
23
24    @Override
25    public List<User> getSearchUser(String queryString, String sort) {
26        String sql = "select * from 'csdlblog'.user where 'User_name' LIKE + " + '%' + queryString + '%' + " order by 'User_name'";
27        try {
28            return jdbcTemplate.query(sql, new BeanPropertyRowMapper(User.class));
29        } catch (Exception e) {
30            return null;
31        }
32    }
33}
```

## - Interface IUserService

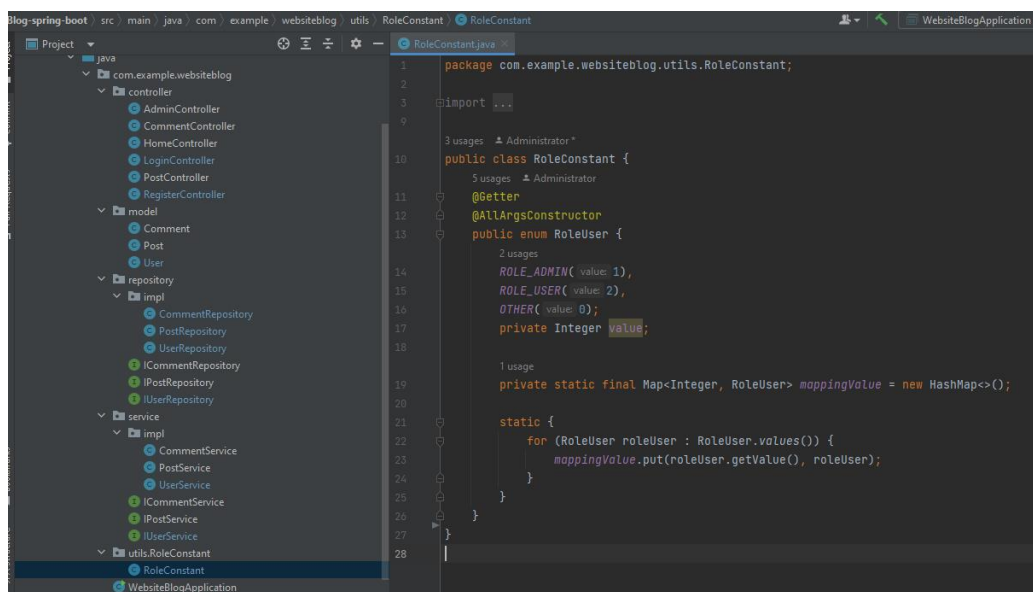


```
1 package com.example.websiteblog.repository;
2
3 import ...
4
5 6 usages 1 implementation 1 dtm0110+1 *
6 public interface IUserRepository {
7     1 usage 1 implementation 1 Administrator
8     public User findByUsername(String username);
9
10    1 usage 1 implementation 1 dtm0110
11    public List<User> getSearchUser(String queryString, String sort);
12
13    1 implementation 1 dtm0110
14    public void delete(Long id);
15
16    1 usage 1 implementation 1 dtm0110
17    public void unban(Long id);
18
19    1 usage 1 implementation 1 dtm0110
20    public User findById(Long id);
21
22    1 implementation 1 Administrator
23    public void save(User user);
24}
```

## - UserService: kế thừa interface IUserService



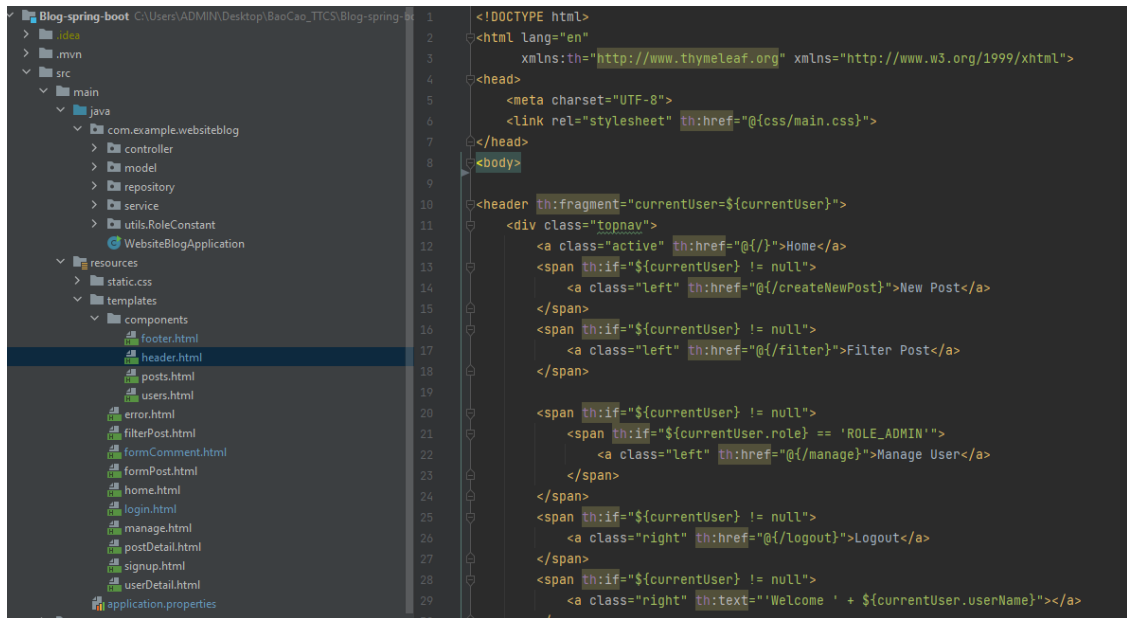
- **RoleConstaint**: phân quyền cho user và admin



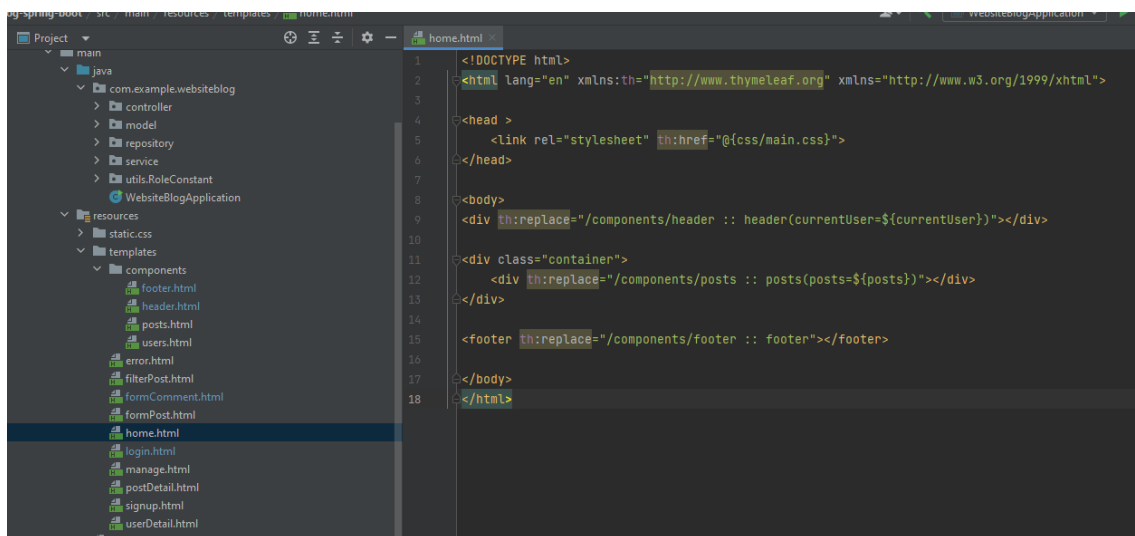
### 5.3.2. Một số hình ảnh demo phần front end

- **Phần components**: file `header.html` : chứa thông tin phần header của trang

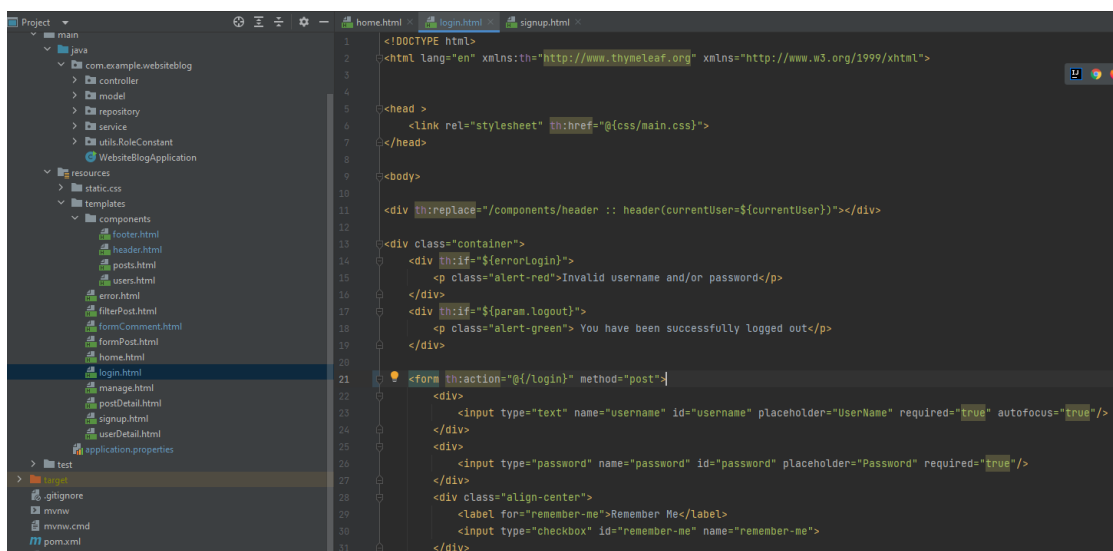




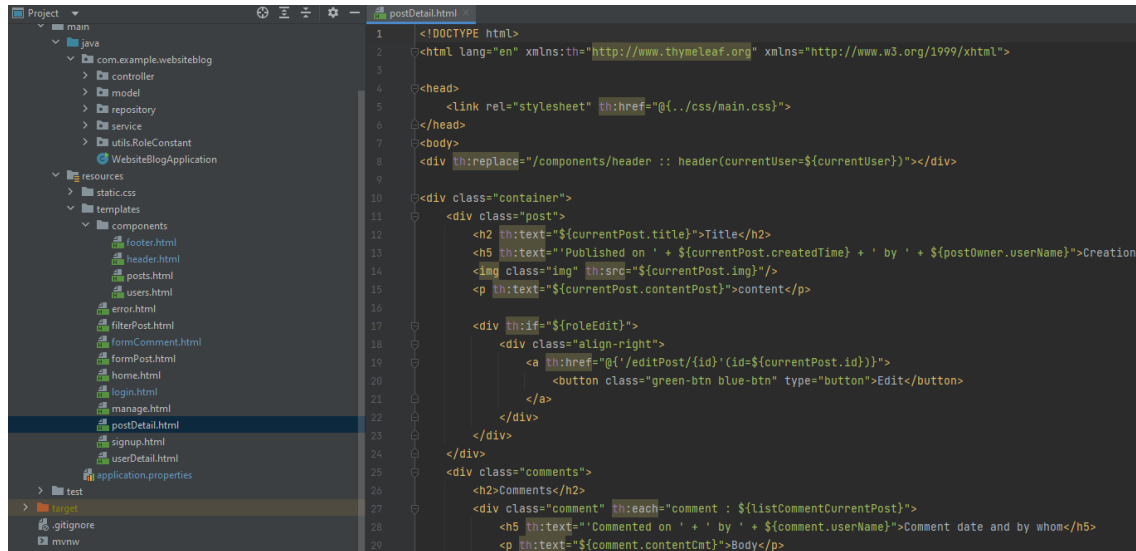
- Trang chủ: file **home.htm**: chứa thư viện của thymeleaf và cấu hình các file thuộc phần **components**



- Trang đăng nhập: file **login.html**

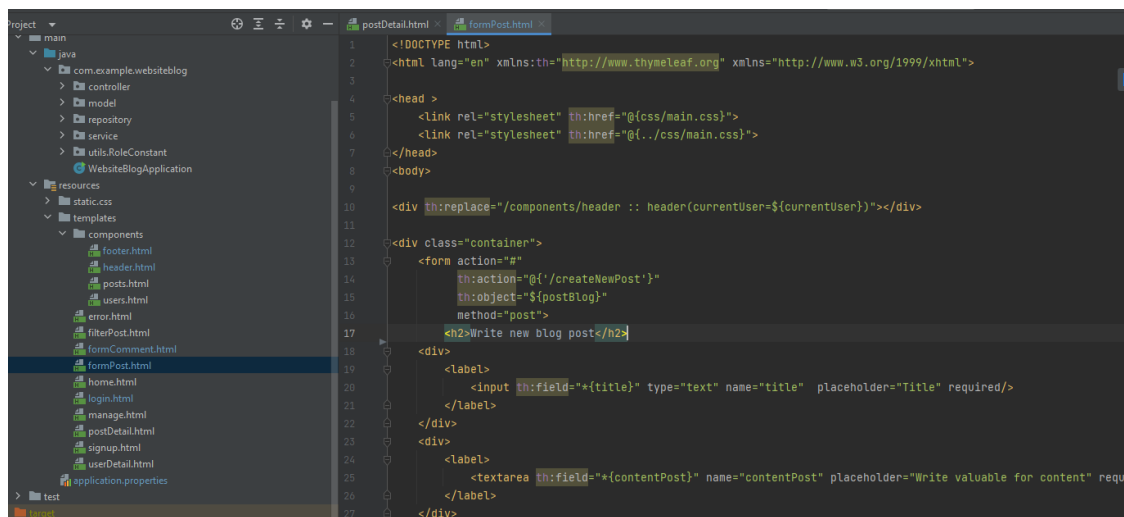


- Trang chủ: phân list các bài viết hiển thị ở trang chủ: file **postDetail.html**



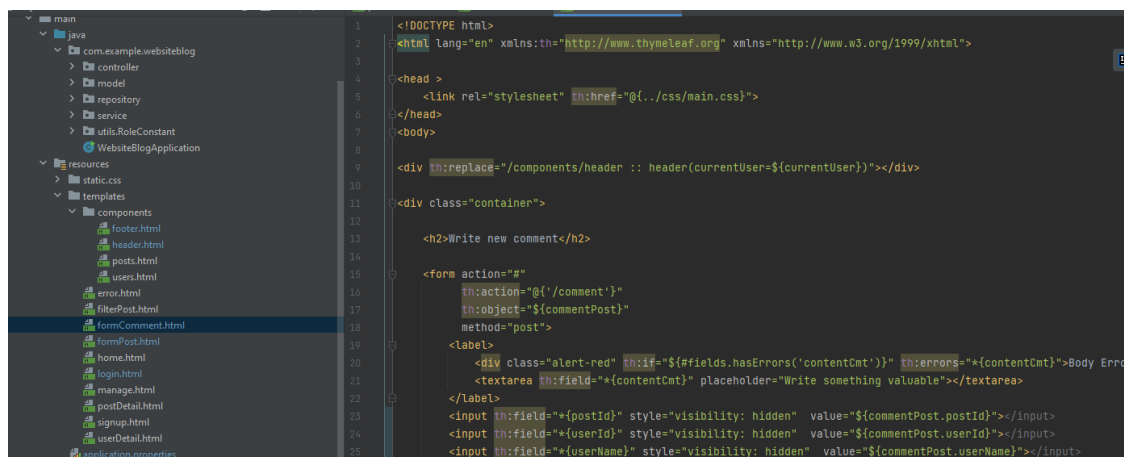
```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org" xmlns="http://www.w3.org/1999/xhtml">
3
4 <head>
5 <link rel="stylesheet" th:href="@{../css/main.css}">
6 </head>
7 <body>
8 <div th:replace="/components/header :: header(currentUser=${currentUser})"></div>
9
10 <div class="container">
11 <div class="post">
12 <h2 th:text="${currentPost.title}">Title</h2>
13 <h5 th:text="'Published on ' + ${currentPost.createdTime} + ' by ' + ${postOwner.userName}">Creation
14 
15 <p th:text="${currentPost.contentPost}">content</p>
16
17 <div th:if="${roleEdit}">
18 <div class="align-right">
19 <a th:href="@{'/editPost/{id}'(id=${currentPost.id})}">
20 <button class="green-btn blue-btn" type="button">Edit</button>
21 </a>
22 </div>
23 </div>
24 </div>
25 <div class="comments">
26 <h2>Comments</h2>
27 <div class="comment" th:each="comment : ${listCommentCurrentPost}">
28 <h5 th:text="'Commented on ' + ' by ' + ${comment.userName}">Comment date and by whom</h5>
29 <p th:text="${comment.contentCmt}">Body</p>
```

- Trang thêm mới bài viết new post: file **formPost.html**



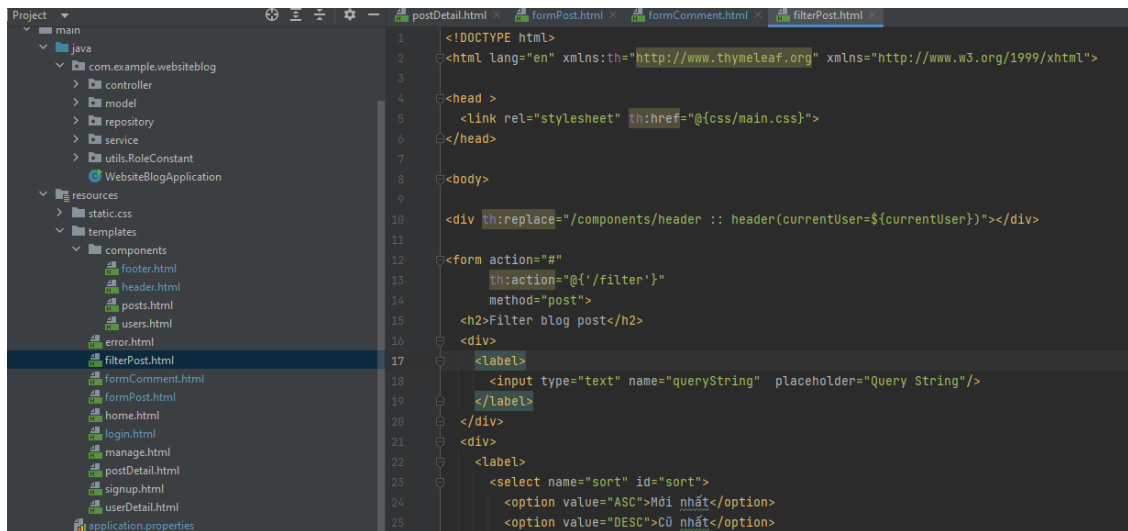
```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org" xmlns="http://www.w3.org/1999/xhtml">
3
4 <head>
5 <link rel="stylesheet" th:href="@{../css/main.css}">
6 <link rel="stylesheet" th:href="@{../css/main.css}">
7 </head>
8 <body>
9
10 <div th:replace="/components/header :: header(currentUser=${currentUser})"></div>
11
12 <div class="container">
13 <form action="#"
14 th:action="@{'/createNewPost'}"
15 th:object="${postBlog}"
16 method="post">
17 <h2>Write new blog post</h2>
18 <div>
19 <input th:field="*{title}" type="text" name="title" placeholder="Title" required/>
20 </div>
21 <div>
22 <input type="text" name="contentPost" placeholder="Write valuable for content" required/>
23 </div>
24 </div>
```

- Trang Comment: file **formComment.html**



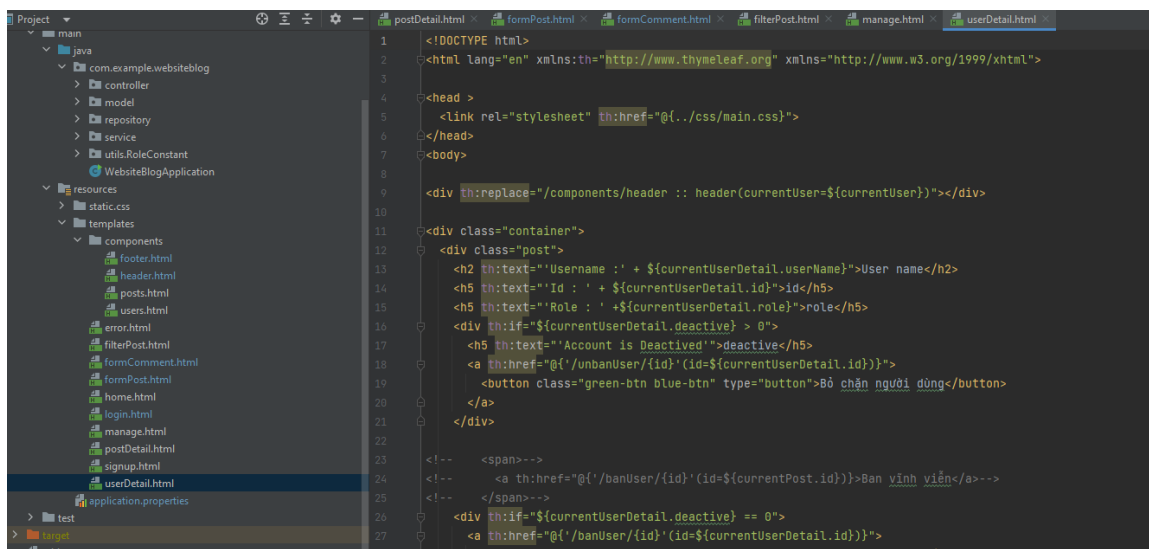
```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org" xmlns="http://www.w3.org/1999/xhtml">
3
4 <head>
5 <link rel="stylesheet" th:href="@{../css/main.css}">
6 </head>
7 <body>
8
9 <div th:replace="/components/header :: header(currentUser=${currentUser})"></div>
10
11 <div class="container">
12 <h2>Write new comment</h2>
13
14 <form action="#"
15 th:action="@{'/comment'}"
16 th:object="${commentPost}"
17 method="post">
18 <div class="alert-red" th:if="${#fields.hasErrors('contentCmt')}">
19 <div class="align-right">
20 <div class="align-right">
21 <div class="align-right">
22 <div class="align-right">
23 <div class="align-right">
24 <div class="align-right">
25 <div class="align-right">
```

- Trang lọc bài viết: **filterPost.html**



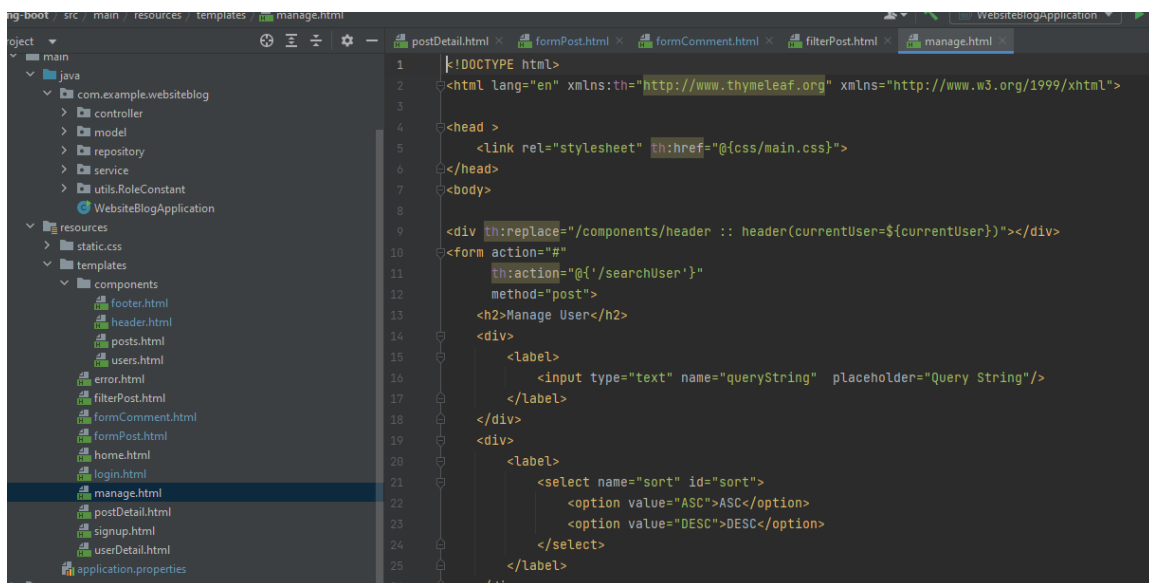
```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org" xmlns="http://www.w3.org/1999/xhtml">
3
4 <head>
5 <link rel="stylesheet" th:href="@{css/main.css}">
6 </head>
7 <body>
8
9
10 <div th:replace="/components/header :: header(currentUser=${currentUser})"></div>
11
12 <form action="#"
13       th:action="@{/filter}"
14       method="post">
15 <h2>Filter blog post</h2>
16 <div>
17 <label>
18 <input type="text" name="queryString" placeholder="Query String"/>
19 </label>
20 </div>
21 <div>
22 <label>
23 <select name="sort" id="sort">
24 <option value="ASC">Mới nhất</option>
25 <option value="DESC">Cũ nhất</option>
```

- Trang quản lý người dùng (dành cho admin): file **userDetail.html**



```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org" xmlns="http://www.w3.org/1999/xhtml">
3
4 <head>
5 <link rel="stylesheet" th:href="@{../css/main.css}">
6 </head>
7 <body>
8
9 <div th:replace="/components/header :: header(currentUser=${currentUser})"></div>
10
11 <div class="container">
12 <div class="post">
13 <h2 th:text="'Username : ' + ${currentUserDetail.userName}">User name</h2>
14 <h5 th:text="'Id : ' + ${currentUserDetail.id}">id</h5>
15 <h5 th:text="'Role : ' + ${currentUserDetail.role}">role</h5>
16 <div th:if="${currentUserDetail.deactive} > 0">
17 <h5 th:text="'Account is Deactivated'">deactive</h5>
18 <a th:href="@{/banUser/{id}}(id=${currentUserDetail.id})">
19 <button class="green-btn blue-btn" type="button">Bỏ chặn người dùng</button>
20 </a>
21 </div>
22
23 <!-- <span>-->
24 <!-- <a th:href="@{/banUser/{id}}(id=${currentUserDetail.id})">Ban vĩnh viễn</a>-->
25 <!-- </span>-->
26 <div th:if="${currentUserDetail.deactive} == 0">
27 <a th:href="@{/banUser/{id}}(id=${currentUserDetail.id})">
```

- Trang lọc người dùng (chỉ dành cho admin): file **manage.html**

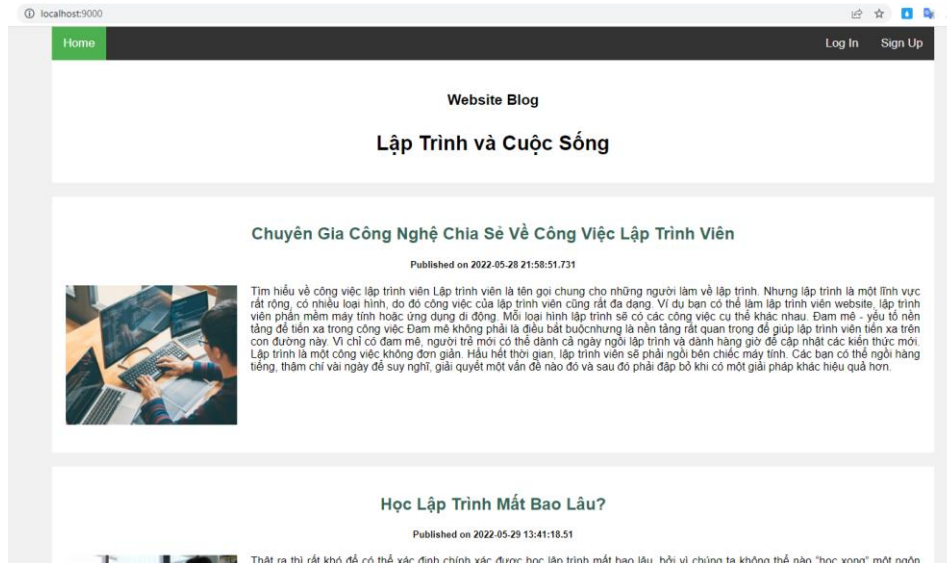


```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org" xmlns="http://www.w3.org/1999/xhtml">
3
4 <head>
5 <link rel="stylesheet" th:href="@{css/main.css}">
6 </head>
7 <body>
8
9 <div th:replace="/components/header :: header(currentUser=${currentUser})"></div>
10 <form action="#"
11       th:action="@{/searchUser}"
12       method="post">
13 <h2>Manage User</h2>
14 <div>
15 <label>
16 <input type="text" name="queryString" placeholder="Query String"/>
17 </label>
18 </div>
19 <div>
20 <label>
21 <select name="sort" id="sort">
22 <option value="ASC">ASC</option>
23 <option value="DESC">DESC</option>
24 </select>
25 </label>
26 </div>
```

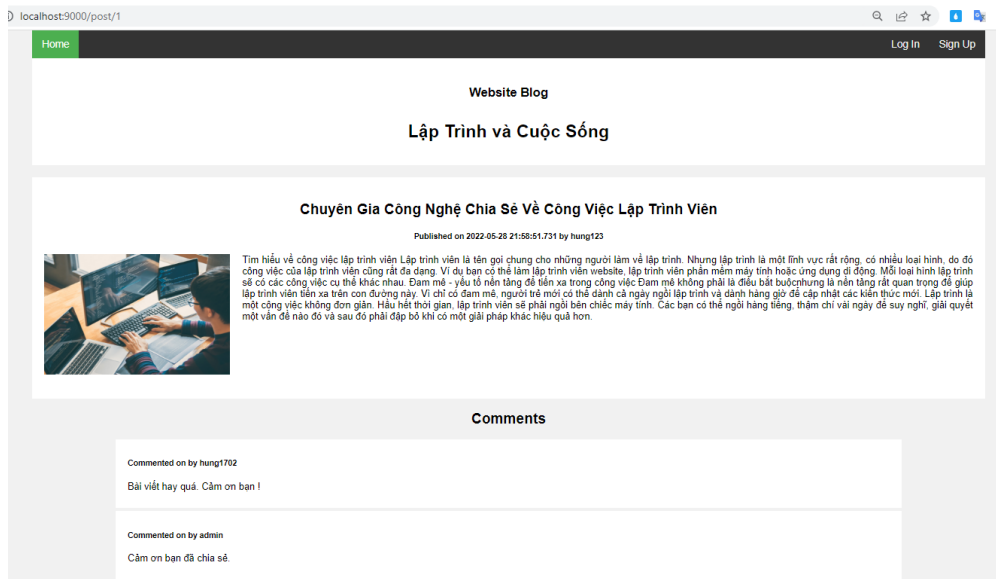
## 5.4. Một số hình ảnh Demo hệ thống website trên localhost:

### Blog Lập trình và Cuộc sống.

+ Khi có link vào hệ thống (chưa đăng nhập):



+ Click vào 1 bài viết ở trang chủ (khi chưa đăng nhập):



+ Trang đăng kí:

The screenshot shows a web browser window with the address bar displaying 'localhost:9000/signup'. The website has a dark header with 'Home', 'Log In', and 'Sign Up' links. The main content area is titled 'Website Blog' and 'Lập Trình và Cuộc Sống'. Below this is a 'Register new user' section with two input fields: 'Username' and 'Password'. A green 'Register' button is positioned below the password field. At the bottom of the page, there is a footer with the text '2022 © Thực tập cơ sở' and 'Directed by P.T.Hung'.

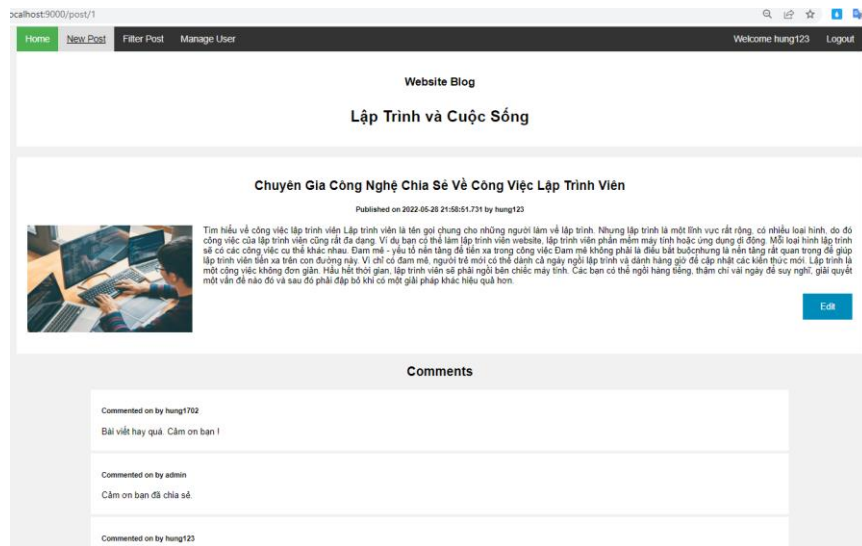
## + Trang đăng nhập

The screenshot shows a web browser window with the address bar displaying 'localhost:9000/login'. The website has a dark header with 'Home', 'Log In', and 'Sign Up' links. The main content area is titled 'Website Blog' and 'Lập Trình và Cuộc Sống'. Below this is a 'Login' section with two input fields: 'Username' (containing 'hung123') and 'Password' (containing '\*\*\*\*'). There is a 'Remember Me' checkbox and a blue 'Login' button. At the bottom of the page, there is a footer with the text '2022 © Thực tập cơ sở'.

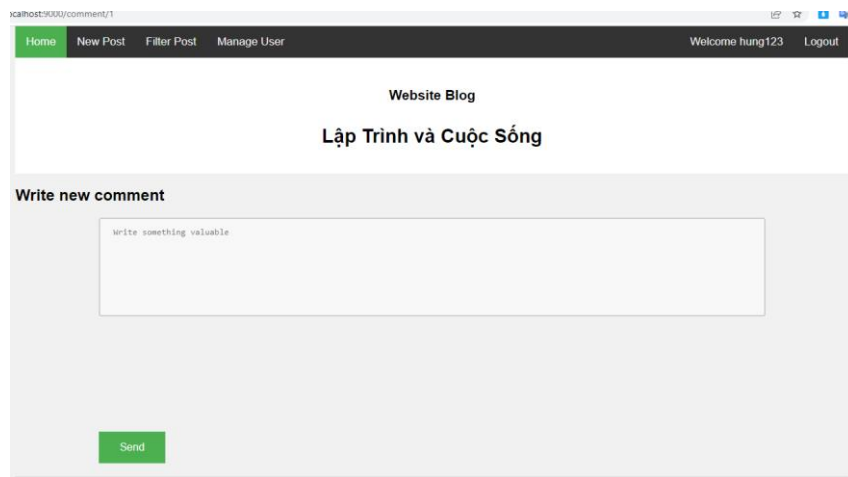
## + Trang home:

The screenshot shows a web browser window with the address bar displaying 'localhost:9000'. The website has a dark header with 'Home', 'New Post', 'Filter Post', and 'Manage User' links. The main content area is titled 'Website Blog' and 'Lập Trình và Cuộc Sống'. Below this is a section titled 'Chuyên Gia Công Nghệ Chia Sẻ Về Công Việc Lập Trình' with a sub-header 'Published on 2022-05-28 21:58:51.731'. There is an image of a person working on a laptop and a paragraph of text. Below this is another section titled 'Học Lập Trình Mất Bao lâu?' with a sub-header 'Published on 2022-05-29 13:41:18.51'. There is an image of a person working on a laptop and a paragraph of text. On the right side of the page, there is a 'logout' button and a 'Save password?' dialog box with 'Username' (hung123) and 'Password' (\*\*\*\*) fields, and 'Save' and 'Never' buttons. Below the dialog box, there is a note: 'Passwords are saved in your Google Account so you can use them on any device'.

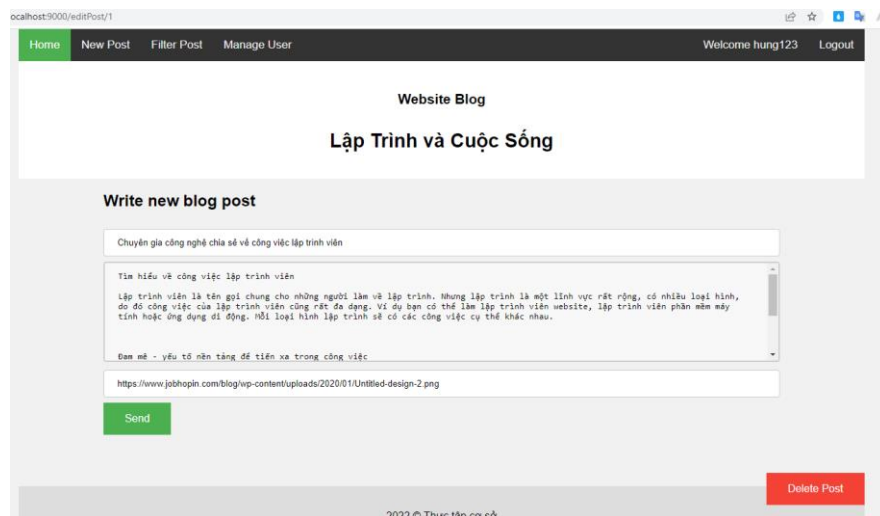
## + Trang thông tin chi tiết bài viết:



+ Trang comment:



+ Trang chỉnh sửa hoặc xóa bài viết:



+Trang New Post:

[Home](#) [New Post](#) [Filter Post](#) [Manage User](#) Welcome hung123 [Logout](#)

**Website Blog**  
**Lập Trình và Cuộc Sống**

**Write new blog post**

## + Trang filter post:

[Home](#) [New Post](#) [Filter Post](#) [Manage User](#) Welcome hung123 [Logout](#)

**Website Blog**  
**Lập Trình và Cuộc Sống**

**Filter blog post**

**Chuyên Gia Công Nghệ Chia Sẻ Về Công Việc Lập Trình Viên**  
Published on 2022-05-28 21:58:51.731



Tìm hiểu về công việc lập trình viên Lập trình viên là tên gọi chung cho những người làm về lập trình. Nhưng lập trình là một lĩnh vực rất rộng, có nhiều loại hình, do đó công việc của lập trình viên cũng rất đa dạng. Ví dụ bạn có thể làm lập trình viên website, lập trình viên phần mềm máy tính hoặc ứng dụng di động. Mỗi loại hình lập trình sẽ có các công việc cụ thể khác nhau. Đảm bảo - yếu tố nền

## + Trang Manage user:

[Home](#) [New Post](#) [Filter Post](#) [Manage User](#) Welcome hung123 [Logout](#)

**Website Blog**  
**Lập Trình và Cuộc Sống**

**Manage User**

**Hung123**

**Hung123456789**

## **Phần 3**

### **Bài tập 3: Máy tính và điều khiển thiết bị**

- *Trình bày kiến trúc và các thành phần cơ bản của máy tính.*
- *Trình bày nguyên lý hoạt động của hệ điều hành*

#### **3.1 Kiến trúc và các thành phần cơ bản của máy tính**

##### **A. Kiến trúc của máy tính**

###### **+ ) Máy tính là gì ?**

Máy tính (Computer) là một thiết bị điện tử được lập trình để nhận dữ liệu đầu vào và xử lý nó với một tập hợp các lệnh gọi là chương trình mục đích cuối cùng là để tạo ra kết quả đầu ra.

Máy tính được thiết kế để thực thi các ứng dụng và cung cấp nhiều giải pháp thông qua các thành phần phần cứng và phần mềm tích hợp. Và máy tính hoạt động được nhờ vào sự trợ giúp của các chương trình và biểu diễn các số thập phân thông qua một chuỗi các chữ số nhị phân.

Kể từ khi ra đời, máy tính đã trở thành một phần không thể thiếu trong cuộc sống hằng ngày. Đóng vai trò như một công cụ phục vụ cho nhu cầu học tập, làm việc và giải trí, máy tính chính “cộng sự” hoàn hảo của con người.

###### **+ ) Hai yếu tố không thể thiếu của máy tính**

Hai yếu tố quan trọng mà bất kỳ máy tính nào cũng phải có, đó chính là: phần cứng và phần mềm.

###### **- *Phần cứng (Computer Hardware)***

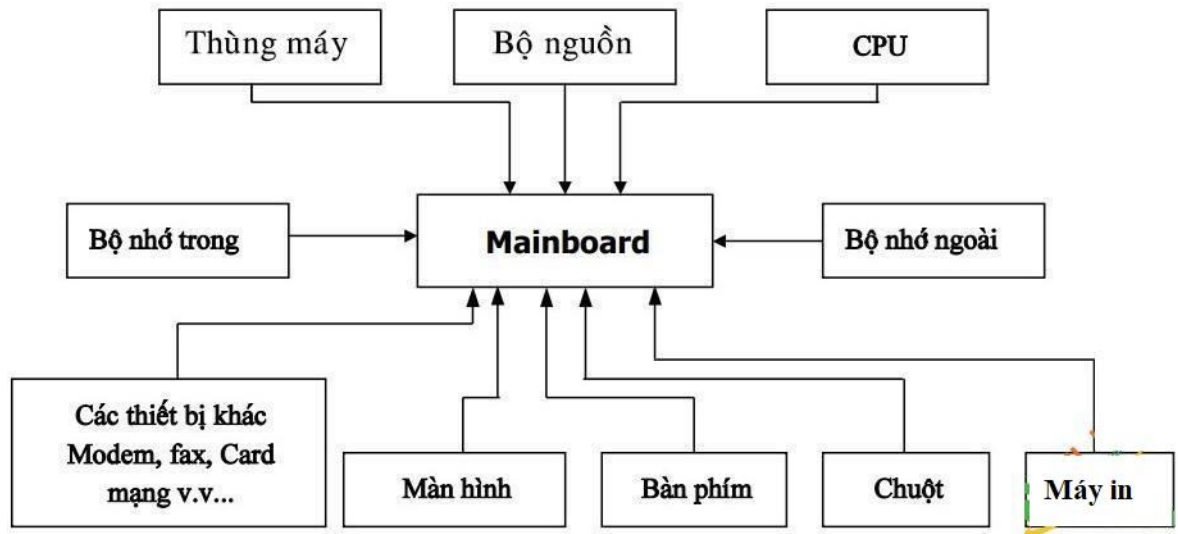
Bao gồm các bộ phận của máy tính có cấu trúc vật lý, nghĩa là người dùng có thể cầm nắm hoặc chạm vào. Đây còn là các thiết bị điện tử chính yếu để cấu tạo nên một chiếc máy tính. Chẳng hạn như: màn hình, bàn phím, chuột, CPU, RAM,...

###### **- *Phần mềm (Computer Software)***

Bao gồm toàn bộ mã lập trình (Programming code) được cài đặt trong ổ cứng máy tính hoặc mainboard. Phần mềm có khả năng ra lệnh để máy tính thực hiện các nhiệm vụ khác nhau.



## **B. Các thành phần cơ bản của máy tính**



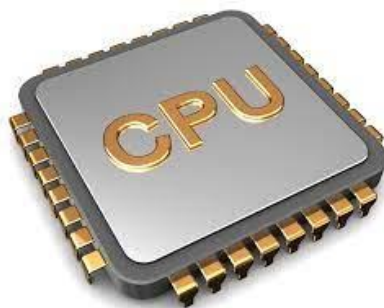
*Sơ đồ các thành phần linh kiện máy tính*

### **+) Các thành phần chính của máy tính**

#### **- Bộ xử lý trung tâm (CPU)**

Bộ xử lý trung tâm CPU (Central Processing Unit) được xem là não bộ của máy tính với nhiệm vụ chính là xử lý các chương trình vi tính, dữ kiện đầu vào của máy tính và xử lý tất cả các lệnh mà CPU nhận được từ phần cứng và phần mềm chạy trên laptop hay máy tính.

CPU là thành phần quan trọng nhất của máy tính, là linh kiện nhỏ nhưng đắt nhất trong máy vi tính.



#### **- Bo mạch chủ (Mainboard)**

Bo mạch chủ hay còn gọi là mainboard/ Motherboard (hay còn được gọi tắt là Mobo hoặc Main) là một bảng mạch in đóng vai trò liên kết các thiết bị thông qua các đầu cắm hoặc dây dẫn phù hợp.



Nhờ có bo mạch chủ, các linh kiện mới có thể hoạt động và phát huy tối đa công năng đạt tới hiệu quả như mong muốn của chiếc máy tính. Đặc biệt, đây còn là thành phần quyết định tuổi thọ của máy tính.

#### **- Bộ xử lý đồ họa (GPU)**

GPU là viết tắt của Bộ xử lý đồ họa, nó là một bộ xử lý chỉ dành riêng cho các hoạt động xử lý đồ họa hoặc tính toán "dấu phẩy động". Một trong những chức năng chính của GPU là giảm tải cho CPU (Bộ xử lý trung tâm), đặc biệt khi chạy các ứng dụng đồ họa chuyên sâu như trò chơi độ phân giải cao hoặc ứng dụng đồ họa 3D.



Bên cạnh trò chơi nặng về xử lý, bộ xử lý đồ họa cũng rất có ích cho những nhà sáng lập mô hình 3D, thiết kế nội thất,... Nói chung, thành phần này có chức năng xử lý tất tần tạt những gì liên quan đến hình ảnh, video hiển thị trên màn hình.

#### **- Bộ nhớ trong (RAM/ROM)**

RAM (**R**andom **A**ccess **M**emory) hay bộ nhớ truy cập ngẫu nhiên của máy tính được sử dụng làm nơi lưu trữ tạm dữ liệu và lệnh thực thi của hệ điều hành và các ứng dụng trước khi ghi chúng lên ổ cứng khi kết thúc phiên làm việc

ROM (**Read-only Memory**) hay còn được gọi là bộ nhớ trong điện thoại hoặc dung lượng lưu trữ. Có chức năng lưu trữ các tệp chương trình, hệ điều hành, thông tin dữ liệu,... trên thiết bị và tồn tại dưới dạng những chip nhớ.



Vai trò của RAM là tạm ghi nhớ những nhiệm vụ cần làm để CPU xử lý nhanh hơn. Thông thường, tốc độ truy xuất trên RAM sẽ cao hơn so với ổ cứng hoặc các thiết bị lưu trữ khác như: đĩa quang, thẻ nhớ.

ROM chính là bộ nhớ trong với đặc trưng sở hữu các dữ liệu đã được cài đặt sẵn và các chương trình giúp thiết bị khởi động ở những lần tiếp theo.

#### - Ổ cứng (HDD, SSD)

Thành phần cuối cùng đóng vai trò cực kỳ quan trọng của máy tính đó chính là ổ cứng. Đây được xem là nơi lưu trữ toàn bộ dữ liệu, phần mềm và hệ điều hành của người dùng. Những loại máy tính hiện đại sử dụng ổ đĩa cứng (HDD) hoặc ổ cứng thể rắn (SSD).



Khác với bộ nhớ tạm RAM, ổ cứng có khả năng ghi nhớ toàn bộ dữ liệu dù bạn bật hay tắt máy tính. Khi máy tính khởi động, hệ điều hành và ứng dụng từ ổ cứng sẽ được chuyển sang bộ nhớ RAM để chạy.

Đơn vị đo dung lượng lưu trữ của ổ cứng là Gigabyte (GB). Thông thường, một ổ cứng có thể chứa từ 500GB đến hơn 1 Terabyte (1.000GB).

**+ ) Các thành phần phụ của máy tính:**

- **Thùng máy:** là nơi để gắn các thành phần của máy tính thành khối như nguồn, Mainboard, Card v.v... có tác dụng bảo vệ máy tính.
- **Bộ nguồn:** là nơi cung cấp hầu hết hệ thống điện cho các thiết bị bên trong máy tính.
- **Bộ nhớ ngoài:** là nơi lưu trữ dữ liệu và chương trình gián tiếp phục vụ cho CPU, bao gồm các loại: đĩa mềm, đĩa cứng, CDROM, v.v... Khi giao tiếp với CPU nó phải qua một thiết bị trung gian (thường là RAM).
- **Màn hình (Monitor):** Là thiết bị đưa thông tin ra giao diện trực tiếp với người dùng. Đây là thiết bị xuất chuẩn của máy vi tính.
- **Bàn phím (Keyboard):** Thiết bị nhập tin vào giao diện trực tiếp với người dùng. Đây là thiết bị nhập chuẩn của máy vi tính.
- **Chuột (Mouse):** Thiết bị điều khiển trỏ giao diện trực tiếp với người sử dụng.
- **Máy in (Printer):** Thiết bị xuất thông tin ra giấy thông dụng nhất.
- **Các thiết bị như Card mạng, Modem, máy Fax,...** phục vụ cho việc lắp đặt mạng máy tính và các chức năng khác.

### **3.2. Nguyên lý hoạt động của hệ điều hành**

#### **A. Hệ điều hành là gì**

- Hệ điều hành là một phần mềm máy tính, nhưng không giống như các phần mềm quen thuộc chúng ta hay dùng hàng ngày như trình duyệt, trình soạn thảo văn bản hay trình biên tập file. Hệ điều hành là tập hợp của các chương trình có chức năng quản lý phần cứng cũng như cung cấp các dịch vụ cần thiết cho các phần mềm khác hoạt động.
- Ngoài ra, hệ điều hành hầu hết sẽ che giấu đi những xử lý tính toán phức tạp, tính toán việc sử dụng tài nguyên cũng như cung cấp các chức năng về cách ly, để tránh tối đa xung đột giữa phần cứng với phần mềm, và tạo một vài lớp bảo vệ tránh tấn công từ bên ngoài (điển hình nhất là Firewall)
- Hệ điều hành có **3 thành phần chính** là

- + **File system** (các tập tin hệ thống )
- + **Scheduler** (bộ lập lịch)
- + **Device driver** (trình điều khiển thiết bị ).
- Có 3 yếu tố cấu thành hệ điều hành bao gồm:
  - + **Abstraction** (yếu tố trừu tượng: gồm có process, thread, file, socket và memory)
  - + **Mechanisms** (yếu tố cơ chế: gồm có create, schedule, open, write, allocate)
  - + **Policies** (yếu tố chính sách: gồm có LRU và EDF)

## **B. Các hoạt động của hệ điều hành**

### **1. Quản lý tiến trình**

- Một tiến trình về cơ bản là một chương trình đang ở trong trạng thái thực thi. Việc thực thi của các tiến trình buộc phải diễn ra theo kiểu tuần tự. Đơn giản là, danh sách các chương trình muốn thực thi sẽ được điền vào một tệp văn bản và khi máy tính hoạt động, nội dung của tệp trở thành thứ tự thực thi các tiến trình tương ứng, nội dung của tệp sẽ được thay đổi theo thời gian.

- Khi một chương trình được nạp vào bộ nhớ và trở thành một tiến trình, nó có thể được chia thành bốn phần - Stack, Heap, Data và Text. Trong đó:

- + **Stack:** Chứa dữ liệu tạm thời, thường chỉ chứa dữ liệu dùng trong thời gian rất ngắn hoặc dùng một lần.
- + **Heap:** Đây là vùng bộ nhớ được cấp phát động cho một tiến trình trong thời gian thực thi.
- + **Data:** Chứa các biến global (biến dùng chung ở cấp toàn bộ chương trình) và static (loại biến dùng chung ở cấp struct).
- + **Text:** Bao gồm hoạt động hiện tại được biểu thị bằng giá trị của Program Counter và nội dung của các thanh ghi trên vi xử lý.

- Khi một tiến trình thực thi, nó có thể đi qua các trạng thái khác nhau. Nhìn chung, một tiến trình có thể có một trong năm trạng thái sau đây tại một thời điểm:

- + **Start:** Trạng thái khi một tiến trình được tạo hoặc được khởi động lần đầu tiên.
- + **Ready:** Tiến trình đang chờ để được chuyển cho vi xử lý.
- + **Running:** Trạng thái thực thi của tiến trình, khi bộ lập lịch Scheduler đã chuyển nó cho vi xử lý, vi xử lý sẽ tiến hành thực hiện các tính toán hoặc chỉ dẫn trên nó.

+ **Wait:** Trạng thái này xảy ra nếu tiến trình cần chờ một tài nguyên để có thể sử dụng, như chờ người dùng nhập liệu hoặc chờ tệp tin đang sử dụng bởi tiến trình khác.

+ **Terminated:** Xảy ra khi một tiến trình đã xử lý xong hoặc bị ngắt bởi yêu cầu từ hệ thống hoặc người dùng, trạng thái này cũng có thể được gọi là Exit.

## 2. Cơ chế lập lịch

- **Scheduling** (lập lịch) là nhiệm vụ của bộ xử lý, nó sẽ tiến hành loại bỏ các tiến trình đang chạy khỏi CPU và lựa chọn một danh sách tiến trình khác trên cơ sở một chiến lược cụ thể. Ngoài các hàng đợi theo trạng thái, hệ điều hành còn duy trì một số hàng đợi như:

+ **Job queue:** Hàng đợi lưu trữ mọi tiến trình đang ở trong hệ thống

+ **Ready queue:** Chứa các tiến trình mới được khởi tạo.

+ **Device queue:** Nếu một tiến trình cần chờ để có thể sử dụng thiết bị I/O thì được đưa vào đây.

- Một số chính sách thường sẽ được sử dụng để lập lịch bao gồm **FIFO**, **Round Robin** hoặc **Priority**,... Theo như trên hình, khi một tiến trình rời khỏi Ready queue để bắt đầu thực thi, nó có thể có 4 khả năng:

+ Request I/O: tiến trình cần xử lý liên quan đến đọc - ghi => Sẽ được chuyển vào Device queue.

+ Timeout: tiến trình đã dùng hết thời gian được cấp phép bởi CPU, sẽ được đưa trở lại vào Ready queue để đợi đến lượt tiếp theo.

+ Fork child: tiến trình tạo ra tiến trình con để tiếp tục xử lý, thường là các tiến trình phức tạp hoặc cần xử lý đồng thời nhiều công việc.

+ Interrupt: tiến trình bị gián đoạn do một vấn đề nào đó đột ngột phát sinh.

## 3. Luồng và xử lý đồng bộ

- Luồng (hay Thread) là một quy trình thực hiện thông qua mã tiến trình, một tiến trình có thể có nhiều luồng thực hiện song song. Nó có bộ đếm chương trình (Program Counter) riêng để theo dõi chỉ dẫn nào sẽ thực hiện tiếp theo. Nó cũng có các thanh ghi hệ thống chứa các biến làm việc hiện tại và một ngăn xếp chứa lịch sử thực hiện các chỉ dẫn.

- Các ưu điểm của luồng bao gồm:

- Giảm thời gian chuyển ngữ cảnh (\*\*Context switch)

- Các tài nguyên được cung cấp đồng thời giữa các luồng của cùng một tiến trình.

- Việc giao tiếp giữa các luồng luôn đạt hiệu quả cao (thay vì là tạo tiến trình con)
- Tiết kiệm chi phí hơn việc dùng đa tiến trình.
- Cho phép tận dụng hiệu quả hơn với vi xử lý hiện đại.

#### 4. Quản lý bộ nhớ

- Quản lý bộ nhớ (hay **Memory Management**) là một chức năng của hệ điều hành, đối tượng được nhắm đến chủ yếu là RAM (Random Access Memory). Các công việc của quản lý bộ nhớ sẽ bao gồm:

- Trao đổi dữ liệu qua lại giữa bộ nhớ RAM và ổ đĩa cứng trong khi một tiến trình được thực hiện.
- Theo dõi từng vị trí cụ thể của bộ nhớ, bất kể nó được cấp cho một tiến trình nào đó đang hoạt động hoặc vùng nhớ đang không được sử dụng.
- Kiểm tra dung lượng bộ nhớ được phân bổ cho các tiến trình.
- Quyết định thời gian được cấp phát bộ nhớ với mỗi tiến trình.
- Theo dõi bất cứ khi nào bộ nhớ được giải phóng hoặc không được phân bổ, và ngay lập tức cập nhật trạng thái tương ứng.
- Tất cả các công đoạn cấp phát, sử dụng, giải phóng hay tiêu hủy đều do bộ phận **Memory management** của hệ điều hành xử lý.

#### 5. Quản lý vào / ra

- Một công việc quan trọng của hệ điều hành là quản lý các thiết bị đầu vào / đầu ra (**Input / Output**) khác nhau, bao gồm chuột, bàn phím, touchpad, ổ đĩa cứng, màn hình, các thiết bị USB, thiết bị kết nối mạng, thiết bị âm thanh, máy in, v.v. Cần thiết phải có hệ thống I / O để nhận các yêu cầu vào / ra dữ liệu của ứng dụng và gửi nó đến thiết bị vật lý, sau đó nhận lại các phản hồi từ thiết bị và gửi đến ứng dụng xử lý tiếp.

- Thiết bị I / O có thể được chia thành hai loại:

- **Block devices** (thiết bị khối): loại thiết bị mà trình điều khiển giao tiếp bằng cách gửi toàn bộ khối dữ liệu. Ví dụ: Ổ đĩa cứng, thiết bị USB Flash, máy ảnh USB ...
- **Character devices** (thiết bị ký tự): Loại thiết bị mà trình điều khiển giao tiếp bằng cách gửi và nhận các ký tự đơn (byte hoặc mã 8 - octa).

#### 6. Công nghệ ảo hóa phần cứng



- Ảo hóa (Virtualization) là công nghệ đặc biệt cho phép chúng ta tạo ra nhiều môi trường mô phỏng hoặc tài nguyên chuyên dụng từ một hệ thống phần cứng vật lý duy nhất (Tức là có thể coi như dùng được nhiều máy tính trên một phần cứng duy nhất).

- Kỹ thuật ảo hóa khá đa dạng, tuy nhiên có thể tóm lại bằng các loại chính sau:

- + Ảo hóa dữ liệu - Data Virtualization
- + Ảo hóa máy tính để bàn - Desktop Virtualization
- + Ảo hóa máy chủ - Server Virtualization
- + Ảo hóa hệ điều hành - Operating System Virtualization
- + Ảo hóa các chức năng mạng - Network Function Virtualization

**+ Tài liệu tham khảo**

- **Kiến trúc và thành phần cơ bản của máy tính:**

- <https://tino.org/vi/thanh-phan-co-ban-cua-may-tinh/?fbclid=IwAR0LQjFUTf-1eoB6vwm5MVmRnyRM6oQD-ojAVOLaBJAi15fH21w8d5wyztM>
- [https://biquyetxaynha.com/hay-neu-thanh-phan-co-ban-cua-may-tinh?fbclid=IwAR1nINOGKOqkO-9lkcsv-eqvILm4G\\_JjrNMJ2akd09iZl1YB7JKN4vapCpo](https://biquyetxaynha.com/hay-neu-thanh-phan-co-ban-cua-may-tinh?fbclid=IwAR1nINOGKOqkO-9lkcsv-eqvILm4G_JjrNMJ2akd09iZl1YB7JKN4vapCpo)

- **Viblo.asia: Hệ điều hành máy tính hoạt động như thế nào:**

- [https://viblo.asia/p/he-dieu-hanh-may-tinh-hoat-dong-nhu-the-nao-phan-1-aWj534BbK6m?fbclid=IwAR3p4SDoBxUUTD6M7Q7rzHRafSCFiKfH4feXV-k0ObJI8SkI\\_vWskvpD8Nic](https://viblo.asia/p/he-dieu-hanh-may-tinh-hoat-dong-nhu-the-nao-phan-1-aWj534BbK6m?fbclid=IwAR3p4SDoBxUUTD6M7Q7rzHRafSCFiKfH4feXV-k0ObJI8SkI_vWskvpD8Nic)
- <https://viblo.asia/p/he-dieu-hanh-may-tinh-hoat-dong-nhu-the-nao-phan-2-4P856kyBKY3?fbclid=IwAR2ZzIz-CFSFgau0vi2QK0eFvFQ9wfc3Pme0GxiJo7L3G5c-fksCsi9UoU>



## Phần: PHỤ LỤC

### Phần Code: Cài đặt 10 thuật toán

#### 1. Sắp xếp chọn.

##### 1.1 Code bằng C++

```
//Sap xep chon
#include<bits/stdc++.h>
using namespace std;
void SapXepChon(int arr[], int n){
    int buoc=1;
    for(int i=0;i<n-1;i++){
        int min=arr[i], vtri=i;
        //tim ptu nho nhat trong mang va lay gatri
        for(int j=i+1;j<n;j++){
            if(arr[j]<min){
                min=arr[j];
                vtri=j;
            }
        }
        //doi cho ptu nho nhat cho ptu hien tai va tiep tục xét tương tự
        int tmp=arr[i];
        arr[i]=arr[vtri];
        arr[vtri]=tmp;
        cout<<"Buoc " << buoc << ": ";
        for(int k=0;k<n;k++){
            cout<<arr[k]<<" ";
        }
        cout<<"\n";
        buoc++;
    }
}

int main(){
    int t;
    cout<<"Nhap so bo test t = ";
    cin>>t;
    while(t--){
        int n;
        cout<<"Nhap so pt mang n = ";
        cin>>n;
        int arr[n+1];
        cout<<"Nhap vào lần lượt các phần tử mảng:\n";
        for(int i=0;i<n;i++){
            cin>>arr[i];
        }
        SapXepChon(arr,n);
    }
    return 0;
}
```

## 1.2 Code bằng **Java**

```
package tt1_sapxepchon;

import java.util.*;

public class TT1_SapXepChon {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Nhap so bo test t = ");
        int t = sc.nextInt();
        while (t-- > 0) {
            System.out.print("Nhap so pt mang n = ");
            int n = sc.nextInt();
            int arr[] = new int[n + 1];
            System.out.print("Nhap vao lan luot cac phan tu mang:\n");
            for (int i = 0; i < n; i++) {
                arr[i] = sc.nextInt();
            }
            SapXepChon(arr, n);
        }
    }

    public static void SapXepChon(int[] arr, int n) {
        int buoc = 1;
        for (int i = 0; i < n - 1; i++) {
            int min = arr[i], vtri = i;
            //tim ptu nho nhat trong mang va lay gatri
            for (int j = i + 1; j < n; j++) {
                if (arr[j] < min) {
                    min = arr[j];
                    vtri = j;
                }
            }
            //doi cho ptu nho nhat cho ptu hien tai va tiep tục xet tuong tu
            int tmp = arr[i];
            arr[i] = arr[vtri];
            arr[vtri] = tmp;
            System.out.print("Buoc " + buoc + ": ");
            for (int k = 0; k < n; k++) {
                System.out.print(arr[k] + " ");
            }
            System.out.println();
            buoc++;
        }
    }
}
```

## 2. Phân tích thừa số Nguyên tố

### 2.1 Code bằng C++

```
//Phan tích thua so nguyen to
#include<bits/stdc++.h>
using namespace std;

void ThuaSoNTTo(long long n){
    cout<<"1 x ";
    //chay tu 2 -> sqrt(n): so i nho nhât ma n chia het se la so ngto
    for(long i=2;i<=sqrt(n);i++){
        int dem=0;
        while(n%i==0){
            dem++;
            n/=i;
        }
        //neu dem != 0 thi in ra cac thua so ngto + so mu (dem)
        if(dem!=0){
            //neu n == 1 thi ko in dau "x" o cuoi va thoat vong for
            if(n==1){
                cout<<i<<"^"<<dem;
                break;
            }
            //neu ko thi in thua so ngto do + so mu(dem) + dau "x";
            else
                cout<<i<<"^"<<dem<<" x ";
        }
    }
    //ket qua n sau vong for = 1 hoac = 1 so ngto
    if(n!=1)
        cout<<n<<"^1";    //chi can in so ngto cuoi cung
    cout<<"\n";
}

int main(){
    cout<<"Nhap so bo test t = ";
    int t;
    cin>>t;
    while(t--){
        cout<<"Nhap 1 so can phan tích thua so nguyen to \nx = ";
        long long n;
        cin>>n;
        if(n<2)
            cout<<n<<" ko the phan tích thua so nguyen to";
        else
            ThuaSoNTTo(n);
    }
    return 0;
}
```

## 2.2 Code bằng Java

```
package tt2_phantichthuasonetojava;

import java.util.*;

public class TT2_PhanTichThuaSoNToJava {

    public static void main(String[] args) {
        // TODO code application logic here
        Scanner sc = new Scanner(System.in);
        System.out.print("Nhap so bo test t = ");
        int t = sc.nextInt();
        while (t-- > 0) {
            System.out.print("Nhap 1 so can phan tich thua so nguyen to \nx = ");

            long n = sc.nextLong();
            if (n < 2)
                System.out.print(n + " ko the phan tich thua so nguyen to");
            else
                ThuaSoNTo(n);
        }
    }

    public static void ThuaSoNTo(long n) {
        System.out.print("1 x ");
        //chay tu 2 -> sqrt(n): so i nho nhat ma n chia het se la so ngto
        for (long i = 2; i <= Math.sqrt(n); i++) {
            int dem = 0;
            while (n % i == 0) {
                dem++;
                n /= i;
            }
            //neu dem != 0 thi in ra cac thua so ngto + so mu (dem)
            if (dem != 0) {
                //neu n == 1 thi ko in dau "x" o cuoi va thoat vong for
                if (n == 1) {
                    System.out.print(i + "^" + dem);
                    break;
                }
                //neu ko thi in thua so ngto do + so mu(dem) + dau "x";
                else
                    System.out.print(i + "^" + dem + " x ");
            }
        }
        //ket qua n sau vong for = 1 hoac = 1 so ngto
        if (n != 1)
            System.out.print(n + "^1");    //chi can in so ngto cuoi cung
    }
}
```

```
        System.out.println();
    }
}
```

### 3. Tìm kiếm nhị phân

#### 3.1 Code bằng C++

```
//Timkiem nhi phan
#include<bits/stdc++.h>
using namespace std;

int TKNhiPhan(int a[], int n, int x){
    int l=0, r=n-1;
    while(l<=r){
        int m=(l+r)/2;
        if(x==a[m])
            return m;
        else if(x<a[m])
            r=m-1;
        else
            l=m+1;
    }
    return -1;
}

int main(){
    cout<<"Nhap so bo test t = ";
    int t;
    cin>>t;
    while(t--){
        cout<<"Nhap so ptu mang \nn = ";
        int n, x;
        cin>>n;
        cout<<"Nhap lan luot cac ptu cua mang:\n";
        int a[n+1];
        for(int i=0;i<n;i++)
            cin>>a[i];
        cout<<"Nhap phan tu can timkiem trong mang: \nx = ";
        cin>>x;

        cout<<"Mang sau khi duoc sap xep:\n";
        sort(a, a + n);
        for(int i=0;i<n;i++)
            cout<<a[i]<<" ";
        cout<<endl;

        int ans = TKNhiPhan(a, n, x);
        if(ans==-1)
            cout<<x<<" ko xuat hien trong mang can tim"<<endl;
        else
```

```
        cout<<x<<" nam o vi tri thu "<<ans+1<<" trong mang can tim"<<endl;
    cout<<endl;
}
return 0;
}
```

### 3.2 Code bằng Python

```
#TT3_TimKiemNhiPhan
def TKNhiPhan(arr, n, x):    #ham tim kiem nhi phan (chat nhi phan)
    l=0
    r=n-1
    while(l<=r):
        m=int((l+r)/2)
        if(x==arr[m]):
            return m
        elif (x<arr[m]):
            r=m-1
        else:
            l=m+1
    return -1

print("Nhap so bo test t = ", end="")
t=int(input())
while(t>0):
    t=t-1
    print("Nhap so ptu mang \nn = ", end="")
    n=int(input())
    print("Nhap lan luot cac ptu cua mang:")
    arr=[]
    for i in range(n):
        arr.append(int(input()))
    print("Nhap phan tu can tim kiem trong mang: \nx = ", end="")
    x=int(input())

    print("Mang sau khi duoc sap xep la:")
    arr.sort()
    for i in range(n):
        print(arr[i], end=" ")
    print()

    ans = TKNhiPhan(arr, n, x)
    if(ans==-1):
        print(x, "ko xuat hien trong mang can tim", sep=" ")
    else:
        print(x,"nam o vi tri thu", ans+1, "trong mang can tim", sep=" ")
```

### 4. hàng đợi ưu tiên (nổi dấy)

## 4.1 Code bằng C++

```
//TT4: Hang doi uu tien (BT noi day)
#include<bits/stdc++.h>
using namespace std;
long long noiday1(int a[],int n){
    priority_queue<int, vector<int>, greater<int> > qu;
    long long i,j,x,x1,x2,ans=0;
    for(i=0;i<n;i++)
        qu.push(a[i]);
    while(qu.size()>1){
        x1=qu.top(); qu.pop();
        x2=qu.top(); qu.pop();
        x=x1+x2;
        ans=ans+x;
        qu.push(x);
    }
    return ans;
}
int main(){
    ios_base::sync_with_stdio(0); cin.tie(0);
    int t;
    cin>>t;
    while(t--){
        int n,i,j;
        cin>>n;
        int a[n];
        for(i=0;i<n;i++)
            cin>>a[i];
        long long x=noiday1(a,n);
        cout<<x<<endl;
    }
    return 0;
}
```

## 4.2 Code bằng Python

```
#TT4_NoiDay_HangDoiUuTien
class PriorityQueue():
    def __init__(self):
        self.queue = []

    def __str__(self):
        return ' '.join([str(i) for i in self.queue])

    def isEmpty(self):    # method kiem tra queue co trong ko: len(queue) = 0
        return len(self.queue) == 0

    def push(self, item): # method chen 1 phan tu vao queue
        self.queue.append(item)
```

```
#trong python: 1 ham vua co the kieu void vua co the kieu return
#-> tích hợp pop() và top() thành delete (mean: vua lay ra ptu theo muc do
uu tien, vua xoa ptu do luon khoi queue)
# method lay ra một phần tử trong queue dựa trên mức độ ưu tiên (sau khi
sắp xếp theo priority queue)
def delete(self):
    try:
        min_val = self.queue[1]          #lay ptu dau tien la min
        for i in range(len(self.queue)):
            if self.queue[i] < self.queue[min_val]:
                min_val = i
        item = self.queue[min_val]
        del self.queue[min_val]
        return item
    except IndexError:
        print()
        exit()

def NoiDay(arr, n):
    myQueue=PriorityQueue()
    ans=0
    for i in range(n):
        myQueue.insert(arr[i])
    while(len(myQueue)>1):
        x1=myQueue.delete()
        x2=myQueue.delete()
        x=x1+x2
        ans=ans+x
        myQueue.push(x)
    return ans

print("Nhap so bo test t = ", end="")
t=int(input())
while(t>0):
    t=t-1
    print("Nhap so ptu mang \nn = ", end="")
    n=int(input())
    arr=[]
    for i in range(n):
        arr.append(int(input()))
    ans=NoiDay(arr, n)
    print("Tong chi phi noi day nho nhat la:", ans)
```

## 5. Sinh tổ hợp Ngược

### 5.1 Code bằng C++

```
//TT5 Sinh tổ hợp ngược
```



```
#include<bits/stdc++.h>
using namespace std;

int a[500];
bool dd;

void HienThi(int a[],int k){
    for(int i=1;i<=k;i++)
        cout<<a[i];
    cout<<" ";
}

void SinhToHopNGUOC(int a[], int n, int k){ //---ham sinh to hop nguoc
    int i=k, tmp=n;
    //vd: 345 -> 5=4+1, 4=3+1, 3 thi lay // 3->2 // 5->5, 4->4
    while(a[i]==a[i-1]+1){
        i--;
        if(i==1)
            break;
    }
    //vd: 123 -> i=1 && a[1]==1 thi dd=false de stop
    if(i==1 && a[1]==1)
        dd=false;
    else{
        a[i]--;
        for(int j=k;j>i;j--){
            a[j]=tmp;
            tmp--;
        }
    }
}

void Solve(int n, int k){
    if(k>n)
        cout<<"n va k nhap vao ko the sinh to hop nguoc!"<<endl;
    else{
        cout<<"Sinh to hop nguoc chap "<<k<<" cua "<<n<<" la:"<<endl;
        a[0]=-1; // loai bo truong hop a[0] = 0
        for(int i=1;i<=k;i++) // vd: n=5, k=3 => khoi tao a=[3, 4, 5]
            a[i]=n-k+i;
        dd=true;
        while(dd){ // ket thuc vong lap khi dd = false
            HienThi(a, k);
            SinhToHopNGUOC(a, n, k);
        }
        cout<<endl;
    }
}

int main(){
```

```
cout<<"Nhap so bo test t = ";
int t;
cin>>t;
while(t--){
    cout<<"Nhap n = ";
    int n, k;
    cin>>n;
    cout<<"Nhap k = ";
    cin>>k;
    Solve(n, k);
}
return 0;
}
```

## 5.2 Code bằng C#

```
using System;
namespace TT5_SinhToHopNguocCS
{
    class Program
    {
        static int[] a = new int[500];
        static bool dd;
        public static void Main(string[] args)
        {
            Console.Write("Nhap so bo test t = ");
            int t = Convert.ToInt32(Console.ReadLine());
            while (t-- > 0)
            {
                Console.Write("Nhap n = ");
                int n = Convert.ToInt32(Console.ReadLine());
                int k = Convert.ToInt32(Console.ReadLine());
                Solve(n, k);
            }
        }

        public static void HienThi(int[] a, int k)
        {
            for (int i = 1; i <= k; i++)
                Console.Write(a[i]);
            Console.Write(" ");
        }

        public static void SinhToHopNGUOC(int[] a, int n, int k)
        {
            int i = k, tmp = n;
            //vd: 345 -> 5=4+1, 4=3+1, 3 thi lay // 3->2 // 5->5, 4->4
            while (a[i] == a[i - 1] + 1)
            {

```

```

        i--;
        if (i == 1)
            break;
    }
    //vd: 123 -> i=1 && a[1]==1 thi dd=false de stop
    if (i == 1 && a[1] == 1)
        dd = false;
    else
    {
        a[i]--;
        for (int j = k; j > i; j--)
        {
            a[j] = tmp;
            tmp--;
        }
    }
}

public static void Solve(int n, int k)
{
    if (k > n)
        Console.WriteLine("n va k nhap vao ko the sinh to hop
nguoc!");
    else
    {
        Console.WriteLine("Sinh to hop nguoc chap {0} cua {1} la:", k,
n);

        a[0] = -1; // loai bo truong hop a[0] = 0
        //vd: n=5, k=3 => khoi tao a=[3, 4, 5]
        for (int i = 1; i <= k; i++)
            a[i] = n - k + i;
        dd = true;
        while (dd)
        {
            //ket thuc vong lap khi dd = false
            HienThi(a, k);
            SinhToHopNGUOC(a, n, k);
        }
        Console.WriteLine();
    }
}
}
}

```

## 6. Sinh hoán vị

### 6.1 Code bằng C++

```

//Sinh hoan vi
#include<bits/stdc++.h>

```

```
using namespace std;
int dd, a[100];
void SinhHoanVi(int a[], int n){
    int i=n-1;
    while(a[i]>a[i+1])
        i--;
    if(i==0)
        dd=0;
    else{
        int j=n;
        while(a[i]>a[j])
            j--;
        swap(a[i], a[j]);
        int l=i+1, r=n;
        while(l<r){
            swap(a[l], a[r]);
            l++; r--;
        }
    }
}

void HienThi(string str, int a[], int n){
    for(int i=1;i<=n;i++)
        cout<<str[a[i]-1]<<" ";
    cout<<endl;
}

int main(){
    cout<<"Nhap so bo test t = ";
    int t;
    cin>>t;
    while(t--){
        cout<<"Nhap 1 chuoí can hoan vi (0 < length < 15)\nstring = ";
        string str;
        cin>>str;
        int n=str.length();
        //cin>>n;
        if(n>0 && n<15){
            cout<<"Chuoí sau khi duoc hoan vi lan luot la:"<<endl;
            for(int i=1;i<=n;i++)
                a[i]=i;
            dd=1;
            while(dd==1){
                HienThi(str,a,n);
                SinhHoanVi(a,n);
            }
        }
        else
            cout<<"Chuoí qua dai, ko the thuc thi"<<endl;
    }
}
```

```
    return 0;
}
```

## 6.2 Code bằng C#

```
using System;
namespace TT6_SinhHoanViCS
{
    class Program
    {
        static int[] a = new int[100];
        static int dd;
        static void Main(string[] args)
        {
            Console.Write("Nhap so bo test t = ");
            int t = Convert.ToInt32(Console.ReadLine());
            while (t-- > 0)
            {
                Console.Write("Nhap 1 chuoai can hoan vi (0 < length < 15)
\nString = ");
                string str = Console.ReadLine();
                int n = str.Length;
                if (n > 0 && n < 15)
                {
                    Console.WriteLine("Chuoai sau khi duoc hoan vi lan luot
la:");
                    for (int i = 1; i <= n; i++)
                        a[i] = i;
                    dd = 1;
                    while (dd == 1)
                    {
                        HienThi(str, a, n);
                        SinhHoanVi(a, n);
                    }
                }
                else
                    Console.WriteLine("Chuoai qua dai, ko the thuc thi");
            }
        }

        public static void HienThi(string str, int[] a, int n)
        {
            for (int i = 1; i <= n; i++)
                Console.Write("{0} ", str[a[i] - 1]);
            Console.WriteLine();
        }

        public static void SinhHoanVi(int[] a, int n)
        {

```

```
        int i = n - 1;
        while (a[i] > a[i + 1])
            i--;
        if (i == 0)
            dd = 0;
        else
        {
            int j = n;
            while (a[i] > a[j])
                j--;
            swap(ref a[i], ref a[j]);
            int l = i + 1, r = n;
            while (l < r)
            {
                swap(ref a[l], ref a[r]);
                l++; r--;
            }
        }
    }

    public static void swap(ref int a, ref int b)
    {
        a = a + b;
        b = a - b;
        a = a - b;
    }
}
```

## 7. Dãy con tăng dài nhất (QHD)

### 7.1 Code bằng C++

```
//TT7: Day con tang dai nhat (QHD)
#include<iostream>
#include<math.h>
#include<algorithm>
using namespace std;
int daycontang(int a[],int n){
    int i,j,f[n];
    for(i=0;i<n;i++)
        f[i]=1;
    for(i=0;i<n;i++){
        for(j=0;j<i;j++){
            if(a[j]<a[i]){
                f[i]=max(f[i],f[j]+1);
            }
        }
    }
}
sort(f,f+n);
```

```
        return f[n-1];
    }
int main(){
    int t;
    cin>>t;
    while(t--){
        int n,i,j;
        cin>>n;
        int a[n];
        for(i=0;i<n;i++){
            cin>>a[i];
        }
        cout<<daycontang(a,n)<<endl;
    }
    return 0;
}
```

## 8. Sàng nguyên tố

### 8.1 Code bằng C++

```
//TT: Sang nguyên tố
#include<bits/stdc++.h>
using namespace std;

void SangNguyenTo(int n){
    bool sang[n+1];
    int i, j;
    for(i=0;i<=n;i++){
        sang[i]=true;
    }
    sang[0]=sang[1]=false;

    for(i=2;i<=sqrt(n);i++){
        if(sang[i]==true){
            for(int j=2; j<=n/i;j++){
                sang[i*j]=false;
            }
        }
    }

    for(i=0;i<=n;i++){
        if(sang[i]==true)
            cout<<i<<" ";
    }
    cout<<endl;
}

int main(){
    cout<<"Nhap so bo test t = ";
    int t;
    cin>>t;
    while(t--){
```

```
        cout<<"Nhap n = ";
        int n;
        cin>>n;
        cout<<"Sang nguyen to tu 0 den "<<n<<" la:"<<endl;
        SangNguyenTo(n);
    }
    return 0;
}
```

## 9.Dijkstra

### 9.1 Code bằng C++

```
//TT9 Dijkstra: giai thuat tim duong di ngan nhat
#include<bits/stdc++.h>
using namespace std;

#define MAX 50
#define TRUE 1
#define FALSE 0
#define VOCUNG 100000000
//so dinh, dinh dau va dinh cuoi
int n, s, t;
//mang danh dau duong di, mang dd khoang cach, dd dinh dc gan nhan
int truoc[MAX], d[MAX], chuaxet[MAX];
//ma tran trong so
int matrix[MAX][MAX];

//ham hien thi duong di
void HienThi(){
    cout<<"Duong di ngan nhat tu "<<(char)(s + 'A' - 1)<<" den "<<(char)(t + 'A' - 1)<<" la:"<<endl;
    cout<<(char)(t + 'A' - 1)<<" <= ";    //inra dinh cuoi thoe kieu char.
    int i = truoc[t];
    while(i != s){
        cout<<(char)(i + 'A' - 1)<<" <= ";    //inra ket qua theo kieu char.
        i = truoc[i];
    }
    cout<<(char)(s + 'A' - 1)<<endl;    //inra dinh dau thoe kieu char.
    cout<<"Do dai duong di la: "<< d[t]<< endl;
    cout<<endl;
}

//giai thuat dijkstra timm duong di ngan nhat
void Dijkstra() {
    int u, minp;
    //khoei tao nhan tam thoi cho cac dinh.
    for (int v = 1; v <= n; v++) {
        d[v] = matrix[s][v];
        truoc[v] = s;
    }
}
```



```

        chuaxet[v] = FALSE;
    }
    truoc[s] = 0;
    d[s] = 0;
    chuaxet[s] = TRUE;
    while (!chuaxet[t]) { //lap mang
        minp = VOCUNG;
        for (int v = 1; v <= n; v++) { //tim dinh u sao cho d[u] là nhỏ nhất
            if ((!chuaxet[v]) && (minp > d[v])) {
                u = v;
                minp = d[v];
            }
        }
        chuaxet[u] = TRUE; //u là dinh có nhãn tạm thời nhỏ nhất
        if (!chuaxet[t]) { //gán nhãn lại cho các dinh.
            for (int v = 1; v <= n; v++) {
                if ((!chuaxet[v]) && (d[u] + matrix[u][v] < d[v])) {
                    d[v] = d[u] + matrix[u][v];
                    truoc[v] = u;
                }
            }
        }
    }
}

int main() {
    cout<<"Nhap so bo test t = ";
    int test;
    cin>>test;
    while(test--){
        cout<<"Nhap so dinh do thi: n = ";
        cin >> n;
        cout<< "Nhap dinh dau do thi: s = ";
        cin >> s;
        cout<< "Nhap dinh cuoi do thi t = ";
        cin >> t;
        cout<<"Nhap cac ptu ma tran cua do thi:"<<endl;
        for (int i = 1; i <= n; i++){
            for (int j = 1; j <= n; j++){
                cin >> matrix[i][j];
                if(matrix[i][j] == 0)
                    matrix[i][j] = VOCUNG;
            }
        }

        Dijkstra();
        HienThi();
    }
    return 0;
}

```

## 10.Quân hậu

### 10.1 Code bằng C++

```
//TT10: Bài toán 8 Quân hậu
#include<bits/stdc++.h>
using namespace std;

int a[100];
//ham in ra ket qua
void HienThi(int n){
    for(int i=1; i<=n; i++)
        cout<<a[i]<<" ";
    cout<<endl;
}
//ham kiểm tra các TH đặt quân hậu có t/m ko, trả về true nếu t/m
bool Check(int c, int r){
    for(int i=1; i<c ;i++)
        if( a[i] == r || abs(a[i]-r) == abs(i-c) )
            return false;
    return true;
}

//ham quy lui quân hậu
void Try(int i,int n){
    for(int j=1; j<=n; j++){ //đặt hậu vào các cột từ 1 đến 8
        if(Check(i, j)){ //nếu có thể đặt thì đặt hậu vào
            a[i] = j;
            if(i==n) //kiểm tra đủ 8 ô thì in ra
                HienThi(n);
            Try(i+1, n);
        }
    }
}

int main(){
    cout<<"Nhập số bộ test t = ";
    int t;
    cin>>t;
    while(t--){
        cout<<"Nhập độ rộng bàn cờ nxn để kiểm tra quân hậu!\nn = ";
        int n; //bàn cờ vua là n = 8
        cin>>n;
        cout<<"Vị trí có thể đặt các quân Hậu theo thứ tự là:"<<endl;
        Try(1,n);
        return 0;
    }
}
```